AD 729848

AD

USAAVLABS TECHNICAL REPORT 70-74B

STABILITY AND CONTROL OF HELICOPTERS IN STEEP APPROACHES

VOLUME II

THE MOSTAB PROGRAM

By

Julian Wolkovitch

John A. Hoffman

May 1971

EUSTIS DIRECTORATE

U. S. ARMY AIR MOBILITY RESEARCH AND DEVELOPMENT LABORATORY

FORT EUSTIS, VIRGINIA

CONTRACT DAAJ02-69-C-0004

MECHANICS RESEARCH, INC.

LOS ANGELES, CALIFORNIA

216

Best
Available
Copy

DISCLAIMERS

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission, to manufacture, use, or sell any patented invention that may in any way be related thereto.

DISPOSITION INSTRUCTIONS

Destroy this report when no longer needed. Do not return it to the originator.

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Mechanics Research, Inc. 9841 Airport Blvd. Los Angeles, California	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP NA

3. REPORT TITLE

STABILITY AND CONTROL OF HELICOPTERS IN STEEP APPROACHES
VOLUME II. THE MOSTAB PROGRAM

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Final Report

5. AUTHOR(S) (First name, middle initial, last name)

Julian Wolkovitch
John A. Hoffman

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
May 1971	214	11

8a. CONTRACT OR GRANT NO. DAAJ02-69-C-0004 b. PROJECT NO. 1F162204A142 c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) USAAVLABS Technical Report 70-74B 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) MRI Report No. 2284-1

10. DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES Volume II of a 4-volume report	12. SPONSORING MILITARY ACTIVITY Eustis Directorate U.S. Army Air Mobility R&D Laboratory Fort Eustis, Virginia

13. ABSTRACT

The general approach used in the MOSTAB modular stability derivative program is described. Details are presented of the aerodynamic representations of modular aircraft elements such as bodies and lifting surfaces, and the rotor dynamic equations are explained. A listing of the MOSTAB program (MOSTAB 'B' version) is given.

DD FORM 1 NOV 65 1473 REPLACES DD FORM 1473, 1 JAN 64, WHICH IS OBSOLETE FOR ARMY USE.

UNCLASSIFIED
Security Classification

UNCLASSIFIED
Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
HELICOPTER STABILITY AND CONTROL						
HELICOPTER AERODYNAMICS						
HELICOPTER STABILITY DERIVATIVES						
HELICOPTER TRANSFER FUNCTIONS						
V/STOL STEEP APPROACH						
VORTEX-RING STATE						
SIKORSKY S-58 STABILITY AND CONTROL						
LOCKHEED AH-56A STABILITY AND CONTROL						
BOEING-VERTOL YHC-1A STABILITY AND CONTROL						
MOSTAB DIGITAL COMPUTER PROGRAM						

UNCLASSIFIED
Security Classification

NOTICE TO USERS

Portions of this document have been judged by the NTIS to be of poor reproduction quality and not fully legible. However, in an effort to make as much information as possible available to the public, the NTIS sells this document with the understanding that if the user is not satisfied, the document may be returned for refund.

If you return this document, please include this notice together with the IBM order card (label) to:

National Technical Information Service
U.S. Department of Commerce
Attn: 952.12
Springfield, Virginia 22151

DEPARTMENT OF THE ARMY
U. S. ARMY AIR MOBILITY RESEARCH & DEVELOPMENT LABORATORY
EUSTIS DIRECTORATE
FORT EUSTIS, VIRGINIA 23604

The report has been reviewed by the Eustis Directorate, U. S. Army Air Mobility Research and Development Laboratory, and is judged to be technically sound.

The primary effort is to examine the behavior of rotary-wing aircraft in steep approaches, from the standpoint of aerodynamics and dynamics, and the resultant effects on human and automatic control.

The report is presented in four volumes. Volume I summarizes the main results of the study. Volume II describes the MOSTAB program. Volume III presents derivatives and transfer functions for the YHC-1A tandem-rotor helicopter and the S-58 single-rotor helicopter. Volume IV presents derivatives and transfer functions for the AH-56A compound helicopter and data on low-altitude turbulence representation.

The program was conducted under the technical management of Mr. William D. Vann, Aeromechanics Division.

Project 1F162204A142
Contract DAAJ02-69-C-0004
USAAVLABS Technical Report 70-74B
May 1971

STABILITY AND CONTROL OF HELICOPTERS IN STEEP APPROACHES

VOLUME II

THE MOSTAB PROGRAM

MRI REPORT NO. 2284-1

By

Julian Wolkovitch
John A. Hoffman

Prepared by

Mechanics Research, Inc.
Los Angeles, California

for

EUSTIS DIRECTORATE
U.S. ARMY AIR MOBILITY RESEARCH AND DEVELOPMENT LABORATORY
FORT EUSTIS, VIRGINIA

ABSTRACT

The general approach used in the MOSTAB modular stability derivative program is described. Details are presented of the aerodynamic representations of modular aircraft elements such as bodies and lifting surfaces, and the rotor dynamic equations are explained. A listing of the MOSTAB program (MOSTAB 'B' version) is given.

FOREWORD

This research was performed by Mechanics Research, Inc. under United States Army Aviation Materiel Laboratories* Contract DAAJ02-69-C-0004, Project 1F162204A142. The AVLABS Project Monitor was Mr. W. D. Vann.

The authors thank Mr. Vann and Mr. Robert P. Smith of AVLABS for their constant encouragement and assistance.

*Redesignated Eustis Directorate, U.S. Army Air Mobility Research and Development Laboratory

TABLE OF CONTENTS

		Page
PART III.	ROTOR ANALYSIS	97
III.1	INTRODUCTION	97
III.2	AXIS SYSTEM DEFINITION	97
III.3	BLADE REFERENCE LINE	99
III.4	INERTIAL ANALYSIS	100
III.5	AERODYNAMIC ANALYSIS	104
III.6	SIMPLIFIED AERODYNAMICS	109
III.7	TOTAL LOADS ON THE BRL	111
III.8	BLADE MOTION EQUATIONS - THE NORMAL MODE METHOD	112
III.9	GENERAL COMMENTS ON THE NORMAL MODE METHOD . .	115
III.10	CALCULATING THE BLADE REFERENCE LINE ANGLES AND COORDINATES	118
III.11	SHAFT LOADS	121
PART IV.	MOSTAB COMPUTATIONAL DETAILS	122
IV.1	INTRODUCTION	122
IV.	LISTING OF MOSTAB 'B' PROGRAM	124
LITERATURE CITED	. . . .	200
DISTRIBUTION	. . . .	201

LIST OF ILLUSTRATIONS

LIST OF SYMBOLS

A_b	The characteristic body area, used in the definition of a body's aerodynamic characteristics
a	Two dimensional lift curve slope of a blade section
a_w	Wing two-dimensional lift curve slope
b_w	LS span
BRL	Blade reference line
C_{DO}, C_{D1}, C_{D2}	Profile drag polar coefficients
C_L, C_D	LS coefficients to lift and drag (referenced to the local relative wind)
C_l, C_m, C_n	Nondimensional moment coefficients. Divisor is $1/2\ \rho V\ A_b \ell_b$
C_l, C_m, C_n	Nondimensional moment coefficients. C_l and C_n are multiplied by $1/2 \rho V^2 S_w b_w$ to get the dimensional quantities, and C_m is multiplied by $1/2 \rho V^2 S_w \bar{c}$
$C_{mo}, C_{m\alpha}$	LS pitching moment coefficients
C_x, C_y, C_z	Nondimensional force coefficients. Divisor $1/2\ \rho V^2 A_b$
C_x, C_y, C_z	Nondimensional force coefficients. Divisor $1/2 \rho V^2 A_b$
c	Rotor blade section chord (generally a function of s).
$\bar{c}$	LS mean aerodynamic chord
cg	Abbreviation for center of gravity. When used as a subscript, cg refers loads and motions to the coordinate system whose origin is located at the cg. Overall vehicle axes and cg axes are parallel.

F_g	First flapping mode generalized force
f_n, f_c	Normal-to-chord and chordwise aerodynamic distributed loading functions.
g	Acceleration of gravity (32.2 ft/sec^2)
h	Altitude
h_x, h_y, h_z	Components of angular momentum of aircraft mass
L_b, M_b, N_b	Aerodynamic body moments referenced to the body's axis system
LS	C_p Lifting Surface (wing or tail)
ℓ_b	Characteristic body length, used to define the moment coefficients
M_g	First flapping mode generalized mass
p, q, r	Rotational velocities of vehicle axes in inertial space
p_x, p_y, p_z	Total BRL distributed loading functions, expressed in rotor coordinates
p_{xa}, p_{ya}, p_{za}	Distributed aerodynamic loading functions, expressed in rotor coordinates
p_{xi}, p_{yi}, p_{zi}	Distributed inertial loading functions, expressed in rotor coordinates
q_a	Dynamic pressure at the reference point of the body
q_a	Dynamic pressure at the LS reference point
R	Rotor radius
S_w	Characteristic area (LS planform area)
s	Radial line coordinate of the BRL: $0 \leq s \leq R$
u, v, w	Translational velocities of vehicle axes in inertial space

$u, v, w,$ p, q, r	Three translational and three rotational inertial velocities of the rotor reference point in hub axis system coordinates
u_A, v_A, w_A p_A, q_A, r_A	Three translational and three rotational airspeeds at the reference point in hub axis system coordinates
u_b, v_b, w_b p_b, q_b, r_b	Translational and rotational airspeeds expressed in axes associated with an aerodynamic body (generally different from overall vehicle coordinates by Euler angles ψ, θ, ϕ)
u_s, u_n, u_c	Spanwise, normal-to-chord, and chordwise airspeeds at a blade section
u_w, v_w, w_w p_w, q_w, r_w	Translational and rotational airspeed components expressed in LS coordinates. These airspeed components apply at the LS reference point. In general, they are different from overall vehicle airspeed components
V	Inertial speed of the aircraft
$V_A \triangleq \frac{d^* \bar{h}}{dt}$	Vector representing the airspeed at a rotor blade section, in rotor coordinates
W	Vehicle gross weight
X_b, Y_b, Z_b	Dimensional forces and moments generated by the body
X_r, Y_r, Z_r L_r, M_r, N_r	Three force and three moment components generated by the LS and referenced to the LS axis system
x, y, z	Coordinates of the BRL referred to rotor axes
x_{cg}, y_{cg}, z_{cg}	Coordinates of the aircraft's center-of-gravity with respect to overall vehicle reference axes
$z_0(s)$	BRL initial shape
$z_1(s)$	BRL first flapping mode shape

α	Body angle of attack $\alpha \triangleq \tan^{-1}\left(\frac{w_b}{u_b}\right)$, radians
α	Angle of attack $\alpha \triangleq \tan^{-1}\left(\frac{w_w}{u_w}\right)$, radians
α_{WCLD}	Angle of attack for zero lift minus angle of attack for minimum profile drag of the two-dimensional airfoil, radians
β	Body sideslip angle $\beta \triangleq \tan^{-1}\left(\frac{v_b}{u_b}\right)$, radians
β	Sideslip angle $\beta \triangleq \tan^{-1}\left(\frac{v_w}{u_w}\right)$, radians
$\beta(t)$	Flapping angle, radians
Γ	Wing dihedral angle (angle between LS xy plane and a wing 1/4 chord line), radians
$\delta_0, \delta_1, \delta_2$	Coefficients of the profile drag polar for a rotor blade section
θ	Angle between a blade section chordline and the shaft normal plane, radians
θ_0, A_{1s}, B_{1s}	Collective pitch, lateral and longitudinal cyclic pitch angles (angles between the control plane and the shaft normal plane), radians
λ_w	Planform taper: LS $\frac{\text{tip chord}}{\text{root chord}}$
ρ	Air density, slug/ft^3, assumed equal to sea level standard density unless otherwise stated
Ψ, Θ, Φ	Euler angles required to rotate vehicle axes from any earth-fixed coordinate system whose z axis coincides with the local gravity vector
ψ	Azimuth angle of blade number 1
$\psi_b, \theta_b, \varphi_b$	Euler angles required to resolve components of vectors expressed in overall vehicle axes to components referred to the local aerodynamic body axes

$\psi_w, \theta_w, \varphi_w$	Euler angles required to resolve components of vectors expressed in overall vehicle axes to components referred to the local LS axes.
Ψ	Rotor speed

Special Mathematical Symbols

$\approx$	Approximately equals
$\triangleq$	Defined equal to
$\lvert x \rvert$	Absolute magnitude of x
x_o	Unperturbed value of x
$\overset{\circ}{x}, \dot{x}$	Derivative of x with respect to time

I. MODULAR STABILITY DERIVATIVE PROGRAM - GENERAL APPROACH

I.1 INTRODUCTION

Chapter V of Volume I of this report describes the MOSTAB modular stability derivative program in a general way, avoiding technical details.

It is emphasized that the version of MOSTAB described here (MOSTAB-B) does not include rotor stall or compressibility as a function of azimuth. It is adequate for the approach conditions considered in this report, but is not suitable for higher speeds, for which later versions of MOSTAB should be employed.

I.2 PROBLEM DEFINITION AND NOTATION

Define a coordinate system x, y, z fixed to the mass of a flying vehicle. The exact location of x, y, z is chosen for convenience during the calculations of aerodynamic forces. One convenient definition for the location of x, y, z on a single rotor helicopter is:

(1) Origin at the intersection of the main rotor shaft and the fuselage waterline.

(2) x axis lying in the vertical plane of symmetry and parallel to the fuselage waterline.

Aircraft are essentially a combination of aerodynamic and inertial elements. These elements may be classified generally into four groups:

(1) Rotating airfoils (lifting rotors, propellers)

(2) Stationary airfoils (wings, empennage surfaces)

(3) Body structures (fuselage, nacelles)

(4) Momentum engines (turbojets, rockets)

Generally, each of these aerodynamic elements produces a force and a moment, which sum (in a vectorial sense) with those forces and moments produced by all other elements. The final sum represents the total load that sustains flight and forces maneuvers.

Now consider an aircraft with N aerodynamic elements. Define a reference point for each element which is convenient for determining loads produced by the element. Locate an axis system x_i, y_i, z_i at each element i, i = 1, 2···N, such that the origin of x_i, y_i, z_i is co-incident with the i'th element's reference point. Fix x_i, y_i, z_i rigidly to the mass of the element reference point, and constrain this coordinate system to remain parallel to the overall vehicle frame of reference, x,y,z.

The force and moment vector generated by each element, i, and applied to the rest of the aircraft can be represented by the six-row column vector f_i. The first three elements of f_i are the force components (in x_i, y_i, z_i coordinates) applied to the aircraft by element i. The last three elements in f_i represent the components of the moment applied to the airframe by element i.

The coordinate system x_i, y_i, z_i has three translational and three rotational velocity components which identify the velocity of x_i, y_i, z_i with respect to inertial space. Define v_{Ii}, a six-element column vector whose first three rows represent translational velocity components and last three rows represent rotational velocity components of the motion of x_i, y_i, z_i in inertial space. In an analogous manner, define v_{Ai} as the velocity of x_i, y_i, z_i with respect to the air in the vicinity of element i. Note that, in general, v_{Ii} and v_{Ai} are different because the air in the vicinity of an aircraft is not still with respect to inertial space. Detailed discussion of these air motions is deferred to a later section.

An aircraft is usually controlled by mechanical reconfiguration of selected aerodynamic elements. Familiar measures of the control configuration are aileron angle, elevator angle, throttle setting, collective pitch setting, etc. To represent these control variables, identify the M-row column vector c. Each element of c represents a control setting. For the present consideration, the order of the elements in c is not relevant. Also, control coordinates which are not varied during a flight case under study (e.g., flaps, throttle) may either be included in c, or may be included elsewhere as physical constants of the system and excluded from c.

The force and moment contributed by each element of an aircraft are generally functions of the local aerodynamic environment, the flight control settings which affect the element, and sometimes the inertial velocity and acceleration of the element. In terms of previously defined notation, this statement can be expressed as a functional mathematical equation:

$$f_i = f_i \left(v_{Ii}, v_{Ai}, \dot{v}_{Ii}, c, K_j, = 1, 2,\right) \qquad (1)$$

$$i = 1, 2, \cdots N$$

where the dot denotes differentiation with respect to time (element by element of v_{Ii}) and K_j, $j = 1, 2, \cdots$ are physical constants of the particular element (wing span, chord, etc.).

Construct the 6NX1 column vectors f, v_I, v_A and $\dot{v}_I$ by simply stacking the 6X1 columns f_i, v_{Ii}, v_{Ai}, and $\dot{v}_{Ii}$, one on top of the other, starting at the top with i = 1. All N equations represented by (1) can then be written as

$$f = f\ (v_I,\ v_A,\ \dot{v}_I,\ c,\ K_j,\ j = 1, 2 \cdots) \tag{2}$$

The force column f represents all the force and moment components produced by all elements of the flight vehicle in x,y,z coordinates. Now define p as the 6X1 column vector whose elements are the three force and three moment components of the total aerodynamic loading on the aircraft. In conventional NACA notation, the elements of p are X,Y,Z,L,M,N. These elements define the load on the aircraft at the origin of x,y,z in x,y,z coordinates.

If the x, y, z, coordinates of each element's reference point are defined, a matrix L can be assembled which relates p to f as follows:

$$p = Lf \tag{3}$$

L is a 6X6N array, and is a function of vehicle geometry only. Thus, p is a function of v_I, v_A, $\dot{v}_I$, c and an unspecified number of physical constants.

Let s represent an aircraft's inertial velocity expressed in x,y,z coordinates. s is a 6X1 column vector made up of three translational and three rotational velocity components. These components have been represented by NACA airplane notation as u,v,w,p,q,r.

If the x,y,z coordinates of the reference point for each vehicle element are defined, a 6NX6 array, G, can be assembled such that

$$v_I = Gs \tag{4}$$

The matrix G is a constant array which depends only on vehicle geometry. Thus,

$$\dot{v}_I = G\ \dot{s} \tag{5}$$

While no proof is given here, it is easy to show that

$$L = G^T$$

It has been stated earlier in this work that the aerodynamic velocity of each vehicle element usually is not the same as its spatial (inertial) velocity, because the air surrounding a vehicle in flight is also moving in inertial space. Neglecting atmospheric wind for the moment, this relative air motion is due to the presence of the vehicle itself. Momentum considerations reveal that aerodynamic forces can be produced by a body with finite dimensions only if that body accelerates the local air mass. Thus, the forces produced by a vehicle element cause the surrounding air to develop velocity components relative to space, and these so-called "interference velocities" impinge not only on the element causing the air motion, but also on other elements of the aircraft. Of course, this velocity interference changes the airloads produced by the other elements from the magnitudes and directions that would be developed if the air mass were still in space. It might be said that interference velocities couple the elements of a flight vehicle aerodynamically.

Let w be the 6NX1 column vector defining the spatial motion of the local air at all of the element reference points. Then

$$v_A = v_I - w \tag{6}$$

The vector w will generally be a function of the airloads produced by all of the vehicle elements, the aerodynamic velocities at all of the elements,and the control settings. Also, certain unsteady aerodynamic effects can cause w to be a function of v_I and $\dot{v}_I$ as well. The functional equation for w can be written as follows:

$$w = w\,(f,\ v_A,\ v_I,\ \dot{v}_I,\ c,\ K_\ell,\ \ell = 1,\ 2\cdots) \tag{7}$$

where K , = 1, 2····are physical constants of the aircraft.

Usually, w is the most difficult quantity to estimate for a flight vehicle. At this point,it must be assumed that some model is available to define the function 7. Analytic, empirical or intuitive models (usually a combination of these three) must be assembled to define w before the dynamics of any flight vehicle can be studied.

The equations presented above represent the general force and moment consideration for the loading of an aircraft in flight. Some form of pilot (human or automatic) produces the column c. Solution of the dynamic equations of motion for the vehicle produces the "velocity state" of the vehicle expressed by the columns s and ṡ. This information, along with the definition

of the vehicle's physical configuration, enables one to compute p, through simultaneous solution of Eqs. (2) - (7). Figure 1 shows these mathematical interrelationships in schematic form.

The entire set of Eqs. (2) - (7) can be represented by the functional expression.

$$p = p\ (s,\ \dot{s},\ c) \qquad (8)$$

This equation is invariably a complicated, nonlinear assemblage of functions actually involving p implicitly. Suppose a solution to (8) is known, of the form

$$p_t = p_t\ (s_t,\ \dot{s}_t,\ c_t) \qquad (9)$$

Let Δp, Δs, $\Delta \dot{s}$, and Δc be small perturbations of p, s, $\dot{s}$, and c from their "trim" or "quiescent" values p_t, s_t, $\dot{s}_t$, and c_t. If the Δ quantities (perturbation quantities) are small, Eq. (8) can be written in the linear form

$$\Delta p = P_s\ \Delta s + P_{\dot{s}}\ \Delta \dot{s} + P_c\ \Delta c \qquad (10)$$

The matrices P_s and $P_{\dot{s}}$ are 6X6 arrays, and P_c is a 6XM array (where M is the number of control variables). In general, the numerical values of P_s, $P_{\dot{s}}$ and P_c are functions of s_t, $\dot{s}_t$ and c_t. Thus, the trim values for s, $\dot{s}$ and c must be specified before numerical values can be assigned to the elements in the rectangular arrays.

The elements of P_s, $P_{\dot{s}}$ and P_c are conventionally called "stability derivatives". For example, in conventional NACA notation, the first element of Δp is the perturbation longitudinal force on the aircraft, ΔX, and the first element of Δs is the perturbed longitudinal spatial velocity, Δu. If all perturbation elements in Δs, $\Delta \dot{s}$ and Δc are zero except for Δu, then

$$\Delta X = P_s\ (1,\ 1)\ \Delta u.$$

Dividing by Δu and taking the limit as $\Delta \to 0$,

$$\lim_{\Delta \to 0} \frac{\Delta X}{\Delta u} = P_s\ (1,\ 1) = \frac{\partial X}{\partial u}$$

The other elements of the rectangular arrays can be defined in an analogous manner, as partial derivatives.

Linear analysis techniques can be used to study the dynamic motions of an aircraft in flight, if the arrays in (1) are numerically defined. (Linear forms of the dynamic equations of motion are easy to derive, and need not be considered here). The "stability derivative problem" is to determine P_s, $P_{\dot{s}}$ and P_c, given Eqs. (2) - (7).

I.3 TRIM

Before the stability derivative matrices can be determined, a "trim" condition must be specified (i.e., the quiescent conditions of velocity state and control, s_t, $\dot{s}_t$ and c_t, must be known). Certain interrelationships among the variables s_t, $\dot{s}_t$ and c_t are stated in defining a "stability derivative case". These interrelationships essentially provide functional equations which can be solved simultaneously with Eqs. (2) - (7) to get the unknown trim columns s_t, $\dot{s}_t$ and c_t. These "interrelationships" that come with the specification of a particular "stability derivative case" will be called "constraints" on the variables in (2) - (7).

To make this concept of constraints clear, consider the following example of a particular stability derivative problem statement.

Find the stability derivatives for H-19 helicopter in steady flight at a constant altitude of 5000 feet with true airspeed (TAS) = 90 knots. The ship is trimmed with zero sideslip angle. Weight = W, cg coordinates = x,y,z with respect to a specific coordinate system.

The statement constrains the variables in Eqs. (2) - (7) by specifying altitude, rate of climb (zero in this case) and airspeed. "Steady" is normally interpreted to mean that $\dot{s}_t = 0$, and all rotational velocities (last three elements of s_t) are zero. Zero sideslip angle constrains the second row in s_t to be zero. Certain physical constants (weight and center-of-gravity position) which vary during a flight, and from flight to flight, are also specified. Enough information must be given in the problem specification so that this information, together with simultaneous solution of Eqs. (2) - (7), will yield all elements of s_t, $\dot{s}_t$ and c_t.

The more detailed presentation concerning trim which follows considers only the cases where $\dot{s}_t = 0$. Although the basic concept of trim does not necessarily require this condition, $\dot{s}_t = 0$ in almost all practical stability derivative problems.

The problem of finding the trim columns s_t and c_t is solved mechanically by a pilot when he trims his aircraft. The pilot's assignment appears in a form similar to the H-19 example given above. He adjusts his flight controls and certain other parameters (e.g., vehicle attitude) available to him until the specification is met. He is essentially solving a set of simultaneous nonlinear equations by iterating on his command over the vehicle until the resulting flight condition converges to his assignment specification (to within certain required accuracy).

The method used by a pilot to trim an aircraft suggests the approach to be taken here for finding the trim columns s_t and c_t. Define the L-row column vector t, whose elements include all of those parameters available for adjustment to trim an aircraft (usually t has six rows), and include certain elements of c and usually information associated with the trimmed altitude of the vehicle in space. For example, the pilot of a pure helicopter adjusts the following six items to trim his ship for level flight with zero sideslip angle.

(1) Collective pitch.

(2) Lateral cyclic pitch.

(3) Longitudinal cyclic pitch.

(4) Tail rotor collective pitch.

(5) Pitch altitude (conventional notation θ).

(6) Roll altitude (conventional notation Φ).

In this case, four elements of c and two vehicle attitude angles are included in t. If the requirement was to trim the vehicle to zero roll angle, sideslip angle, β, would be included in t in lieu of Φ.

The trim control column, c_t, is generally a function of t:

$$c_t = c\ (t,\ \text{known constraints, known constants}) \qquad (11)$$

The six trim variables listed above indicate a 1-to-1 relationship between certain elements of c and the corresponding elements of t. This is not necessarily always the case. For example, longitudinal stick position may be defined as one element of t. In most helicopters, longitudinal stick position affects both lateral and longitudinal cyclic pitch angles (i.e., two

elements of c). Nonlinear expressions may relate elements of c to elements of t (e.g., an aircraft may include nonlinear mechanical couplings between pilot inputs and control variables).

The column s_t can be calculated from the constraints of the problem and the specification of t:

$$s_t = s_t \text{ (t, known constraints)} \tag{12}$$

For example, the numerical values of the elements in the last three rows of s will usually be specified by the problem statement. The second element (sideslip velocity) will either be zero, or it will be included in t. The usual specification of airspeed and flight path angle will allow calculation of the first and third row elements of s_t from the altitude rate equation. The solution will normally be a function of the vehicle attitude variables in t.

From the definition of t outlined above, one sees that $c_t(t)$ and $s_t(t)$ can be directly calculated as soon as a numerical value for t is available. With the statement that $\dot{s}_t = 0$, simultaneous solution of Eqs. (2) - (7) will eventually lead to the solution of p. This process is represented by the functional expression,

$$p = p \text{ (t, stability derivative problem constraints, physical constants)} \tag{13}$$

The unique value of t required to trim an aircraft must be determined from Eq. (12) and the stability derivative problem statement. The problem statement must require a specific value for p. This "required" p column can be equated to the p column shown in 13), to yield an L-row vector equation with t as its only unknown. Remember that t itself has L-rows. This process produces L (generally nonlinear) equations in L unknowns (elements of t).

Let r be the "required" trim value for p; r will generally come from the six equations of motion for the aircraft, as constrained, by the stability derivative problem statement. Since t usually contains elements related to vehicle attitude, and since the equations of motion for a flying vehicle contain terms dependent on attitude, r is generally a function of t.

$$r = r \text{ (t, problem constraints, constants)} \tag{14}$$

The trimming problem, in terms of the functional expressions now available, can be stated very simply: Find t such that p(t) = r(t).

Numerical Solution for the Trimmed Condition (Trim Search Iteration)

In previous sections, certain functional expressions were presented. These expressions are summarized below for convenience. They retain their original statement numbers. Indication that some of the expressions rely on constants known to the trim problem is dropped. The vector $\dot{v}_I$ is also dropped, since it will be zero for all trim cases considered using MOSTAB. The subscript, t, indicating the trim condition in some of the previous expressions of these equations, is dropped.

$$f = f(v_I, v_A, c) \tag{2}$$

$$p = Lf \tag{3}$$

$$v_I = Gs \tag{4}$$

$$v_A = v_I - w \tag{6}$$

$$w = w(f, v_A, v_I, c) \tag{7}$$

$$c = c(t) \tag{11}$$

$$s = s(t) \tag{12}$$

$$r = r(t) \tag{14}$$

It is assumed that explicit relationships of the forms shown above are available to the trim problem (i.e., a numerical value for the left hand side of each expression can be determined if numerical values for the variables in the arguments are defined).

Eqs. (3), (4) and (6) are always linear (constant L and G). The others are generally nonlinear. Note that, even if t is known, p cannot be explicitly determined because of the nonlinear involvement of f and w in Eqs. (2) - (7).

Estimate the value of t that will trim the aircraft within the framework of the constraints given to the stability derivative problem. Also estimate w. Denote these estimated columns as t_e and w_e. Using these estimates, calculate the following quantities:

$$c_o = c\,(t_e) \tag{15}$$

$$s_o = s\,(t_e) \tag{16}$$

$$v_{Io} = Gs_o \tag{17}$$

$$v_{Ao} = v_{Io} - w_e \tag{18}$$

$$f_o = f\,(v_{Io},\, v_{Ao},\, c_o) \tag{19}$$

$$w_o = w\,(f_o,\, v_{Ao},\, v_{Io},\, c_o) \tag{20}$$

$$r_o = r\,(t_e) \tag{21}$$

$$t_o = t_e \tag{22}$$

Let the difference between the correct value (the sought trim solution value) of each variable and the subzero value be denoted by Δ (variable). For example, if f_t is the true value of f for the trimmed aircraft, then

$$\Delta f = f_t - f_o$$

and so on for all of the other variables. If the Δ quantities are small, the nonlinear Eqs. (2), (7), (11), (12), and (14) might be suitably represented in the following linearized forms:

$$2)_L \quad f = f_o + F_{VI}\,\Delta v_I + F_{VA}\,\Delta v_A + F_C\,\Delta c$$

$$7)_L \quad w = w_o + W_F\,\Delta f + W_{VA}\,\Delta v_A + W_{VI}\,\Delta v_I + W_C\,\Delta c$$

$$11)_L \quad c = c_o + C_T\,\Delta t$$

$$12)_L \quad s = s_o + S_T\,\Delta t$$

$$14)_L \quad r = r_o + R_T\,\Delta t$$

The rectangular arrays, shown above as upper case letters, can be assumed constants if the Δ quantities are small. These arrays are functions of the original estimates t_e and w_e, and are easily calculated from Eqs. (2), (7), (11), (12), and (14) using a digital computer.

The procedure for finding a trim solution now becomes that of solving a set of linear equations for Δt and Δw.

Combining Eqs. (4), (6) and $(12)_L$,

$$v_A = G (s_o + S_T \Delta t) - (w_o + \Delta w) \tag{23}$$

Subtracting (18) from (23), using (17) to eliminate v_{Io}, one gets

$$\Delta v_A = GS_T \Delta t - (w_o - w_e) - \Delta w \tag{24}$$

Eqs. (4), $(2)_L$, $(7)_L$, $(11)_L$, $(12)_L$, and (24) can be combined to get Δv_A as a function of $(w_o - w_e)$ and Δt. (Note that $(w_o - w_e)$ is not the same as Δw. This quantity is known at this point, since w_o was computed as Eq. 20.) The resulting equation for Δv_A is*

$$\Delta v_A = V_T \Delta t - V_W (w_o - w_e) \tag{25}$$

where

$$V_W = (1 + W_F F_{VA} + W_{VA})^{-1} \tag{26}$$

and

$$V_T = V_W \left[(1 - W_F F_{VI} - W_{VI}) GS_T - (W_F F_C + W_C) C_T\right] \tag{27}$$

Noting that

$$\Delta v_I = G S_T \Delta t \tag{28}$$

from Eqs. (4) and $(12)_L$, Eqs. $(2)_L$, $(11)_L$, (25), and (28) can be combined to yield

$$f = f_o + F_T \Delta t - F_W (w_o - w_e) \tag{29}$$

*The notation 1 refers to the "unit" array: elements with equal subscripts are unity and all others are zero.

where

$$F_T = F_{VI}\, GS_T + F_{VA}\, V_T + F_C\, C_T \tag{30}$$

and

$$F_W = F_{VA}\, V_W \tag{31}$$

Combining (29) and (3),

$$p = Lf_o + LF_T\, \Delta t - LF_W\, (w_o - w_e) \tag{32}$$

Equating $(14)_L$ to (32) - the requirement for trim - and solving for Δt,

$$\Delta t = (R_T - LF_T)^{-1} \left[Lf_o - r_o - LF_W\, (w_o - w_e) \right] \tag{33}$$

This value for Δt can be substituted into (25) to get Δv_A and into(29) to get Δf. These results can be substituted into Eqs. $(7)_L$, $(11)_L$ and (28) to get Δw:

At this point, new estimates on t and w can be made:

$$t_e \big|_{new} = t_o + \Delta t \tag{34}$$

$$w_e \big|_{new} = w_o + \Delta w \tag{35}$$

These new estimates on t and w can be used to repeat the process again. The cycling can occur as often as time permits, until the differences between the old and new estimates for t_e and w_e are within some acceptably small values. The chosen "acceptability limits" should be based on the physical dimensions of the elements of w and t. One test procedure could be

$$\text{(acceptable tolerance on t)} \geq \sum_{\text{all elements of t}} \left| t_{e\ new} - t_{e\ old} \right| \tag{36}$$

$$\text{(acceptable tolerance on w)} \geq \sum_{\text{all elements of w}} \left| w_{e\ new} - w_{e\ old} \right|$$

Discussion of the Trim Search Iteration

While considering any numerical iteration process, the question of convergence arises. This question will not be treated with any mathematical approach here. The comments to be extended are quite intuitive.

Whether or not the trim solution method outlined in the previous section converges to a solution seems to depend on two factors:

(a) The nature of the specific nonlinear functions used to represent the aircraft's characteristics.

(b) The correctness of the original estimates, t_e and w_e.

Certainly, the degree of nonlinearity characterized by the aircraft's aerodynamic functions will affect the rapidity of convergence, or indeed whether convergence occurs at all. If all of the aerodynamic expressions are completely linear, convergence to the exact trim solution will occur with only one cycle. On the other hand, if the problem statement assigns a trim condition within a flight regime unattainable by the aircraft, no trim solution can exist. Hopefully, the iteration search will indicate this by failing to find a solution.

For those "difficult" regions in which a trim solution does exist, but may not be found by the iteration process, a more sophisticated iteration method may be required. One such method may be simply to add some of the higher order (nonlinear) terms to be the "first term only" Taylor expansions $(2)_L$, $(7)_L$, $(11)_L$, and $(14)_L$. The additional complexity of this approach may not be justified, if the solution can be found by invoking engineering judgement to produce better initial estimates. The first-order convergence method proposed here always must converge on a solution if the initial guess is close enough.

The idea that the accuracy of the initial estimates might affect convergence provokes one to consider a method for "sneaking up" on that "difficult" solution. This approach would proceed as follows:

(a) Begin by finding a solution for trim in a nearby region to that in which convergence has been found difficult.

(b) Progressively change the problem statement toward that statement representing the difficult region. For each step, compute a trim solution and use this solution as the initial estimate for the solution of the next step.

If the convergence technique becomes too time consuming, certain simple changes in approach might be attempted to shorten the computational time required. For example: the rectangular arrays in the linearized Eqs. $(2)_L$, $(7)_L$, $(11)_L$, $(12)_L$, and (14) are recomputed during each iteration cycle for the iteration approach suggested in the previous section. It may not be necessary to do this every cycle. Computing new arrays every M cycles (M > 1) may save time but will not affect the ability of the method to converge on a solution.

To see how this abbreviated method works consider Figure 2. Let y_s be the required solution-value for y. The problem is to find x_s. Estimate x_e as the solution. Compute y_o and the gradient (slope), s. (The slope, s, in this example, is analogous to the linear arrays in vector expressions $(2)_L$, $(7)_L$, etc.) Using s, and the known error $y_s - y_o$, determine x_1 as the next proposed solution. Continue this process, but use the same slope value each time. As one sees from this figure,the iteration is converging on the solution, even though s is held constant.

Whether or not this abbreviated method shortens convergence time depends on the complexity of the aerodynamic expressions - particularly f and w. The linear arrays are computed, numerically, column by column. Every time a column is generated in F_c, for example, values for the elements in f must be calculated. If this calculation is even moderately time-consuming, finding the numerical values for the arrays will be very time-consuming. In this case, the iteration process can probably be accelerated using the abbreviated method.

I.4 STABILITY DERIVATIVE CALCULATION

The linearized expression derived in the trim solution can be used to generate the stability derivative matrices. The terms $F_{\dot{V}I} \Delta\dot{v}_I$ and $W_{\dot{V}I} \Delta\dot{v}_I$ must be added to Eqs. $(2)_L$ and $(7)_L$, respectively, to account for the dependency of f and w on $\dot{v}_I$. Recollect that these terms were not required for the trim case, because non-zero $\dot{s}$ trim cases were not considered.

Combining Eqs. (3), (4), (6), $(2)_L$, and $(7)_L$ with the added $\Delta\dot{s}$ terms, the following expression is derived:

$$\Delta p = \left\{ L \left[F_{VI} + F_{VA} V_W (1 - W_F F_{VI} - W_{VI}) \right] G \right\} \Delta s$$
$$+ \left\{ L \left[F_{\dot{V}I} - F_{VA} V_W (W_F F_{\dot{V}I} + W_{\dot{V}I}) \right] G \right\} \Delta \dot{s}$$
$$+ \left\{ L \left[F_C - F_{VA} V_W (W_F F_C + W_C) \right] \right\} \Delta c$$

Comparing (38) to (10) shows that the factors in braces are the required stability derivative arrays.

I.5 ROTORS WITH FLEXIBLE BLADES

In previous sections, it has been assumed that the forces generated by all N vehicle components can be represented by a model having the form

$$f_i = f_i \left(v_{Ii}, v_{Ai}, \dot{v}_{Ii}, c, K_j, j = 1, 2, \ldots\ldots \right)$$
$$i = 1, 2, \ldots\ldots N \qquad (1)$$

Given the columns v_{Ii}, v_{Ai}, $\dot{v}_{Ii}$, and c (along with the physical constants), Eq. (1) can be used to calculate f_i. This equation is not intended to represent dynamic interfacing between f_i and its functional argument. Eq. (1) is purely a static relationship.

Most vehicle elements have independent dynamic characteristics. Lifting surfaces and bodies have structural vibration modes. Engines have lags and high frequency oscillatory characteristics. Usually, these dynamic effects can either be neglected because they involve frequency ranges far removed from those of interest for flight dynamics considerations, or they can be included in some simple peripheral manner (e.g., a simple lag on throttle command might be used to represent engine dynamics). In the special case of rotors with flexible blades, the dynamics of the blades must be considered, because blade motion has an extremely important influence on flight dynamics.

To say that blade dynamics have an important influence on flight dynamics does not imply that a static function (1) cannot be defined for a rotor with flexible blades. The function (1) is called a "quasi-static" representation when applied to a rotor with flexible blades. In such a quasi-static representation, f_i

is still defined as a static function of the argument (i.e., no dynamic characteristics are included in the transfer function $f_i = f_i$ (arg)). However, the dynamic motions of the blades are included in that they affect the actual static value of f_i in a substantial manner.

In reality, time should be included in the argument of (1) when applied to the flexible rotor. However, such dynamic effects usually have little influence on vehicle flight dynamics. If special rotor structural dynamics are being studied (i.e., flutter, vibrations and some mechanical stability augmentation schemes), the quasi-static assumption (1) is not appropriate, and rotor dynamics must be considered.

Blade motion equations, written in a coordinate system fixed to the rotation hub, generally appear in the following form:

$$\ddot{\beta}_j = g_j \quad j = 1, 2 \cdots \infty, \tag{39}$$

where β_j is the coordinate of the blade's j'th degree-of-freedom. (Flexible blades have an infinite number of degrees-of-freedom, as expressed by Eq. (3). The driving function g_j usually contains coordinates of all blade degrees-of-freedom, and time functions known to the blade motion problem. This functional dependency of g_j can be expressed in the form

$$g_j = g_j\,(\beta_\eta, \eta = 1, 2, 3, \cdots \infty, \dot{\beta}_\eta, \eta = 1, 2, 3, \cdots \infty, \psi, t, K_\ell, \ell = 1, 2 \cdots) \tag{40}$$

where ψ is rotor blade azimuth position and K_ℓ are physical constants associated with the blade. Note that ψ is a function of time, as are the columns v_{Ii}, v_{Ai}, ϑ_{Ii}, and c.

Each rotor blade applies a force and a moment to the rotating hub. This phenomena can be represented in the form

$$f_{r_k} = f_{r_k}\,(\beta_\eta, \eta = 1, 2, 3 \cdots \infty, \dot{\beta}_\eta, \eta = 1, 2, 3, \cdots \infty, \psi, t, K_r, r = 1, 2 \cdots) \tag{41}$$

$$k = 1, 2 \cdots \text{total number of rotor blades.}$$

f_{r_k} is a 6X1 column vector containing three force and three moment components which represent the loads applied by the k'th blade to the rotating hub. These components are stated in terms of some convenient frame-of-reference fixed to the rotating hub.

The rotating forces, f_{r_k}, can be summed over all the rotor blades, transformed to the nonrotating airframe through some ψ-dependent transformation matrix, and time averaged. The result is f_i:

$$f_i = \frac{1}{T}\int_0^T \left[R(\psi) \sum_{\substack{\text{all} \\ \text{blades}}} f_{r_k}\right] dt \tag{42}$$

Although v_{Ii}, v_{Ai}, $\dot{v}_{Ii}$, and c are generally functions of time, they can be considered constant while deriving the quasi-static rotor model (1). In this special case, the t can be removed from the arguments of Eqs. (40) and (41). Also, since the rotor is being treated as a quasi-static entity, the degrees-of-freedom associated with the blades can be assumed periodic over the period $2\pi/\Omega$, where Ω is the constant rotor spin rate.

$$\psi = \Omega t \tag{43}$$

Because t has been removed from the arguments in Eqs. (40) and (41), and because the blade degrees-of-freedom move periodically with $2\pi/\Omega$ (which means over the azimuth angle $0 \leq \psi < 2\pi$), Eq. (42) can be expressed in the azimuth-average form:

$$f_i = \frac{b}{2\pi}\int_0^{2\pi} R(\psi)\, f_r\, d\psi \tag{44}$$

where b is the total number of blades and f_r is f_{r_k} for any blade. Form (44) is possible because the motion is periodic. Thus, all blades move in exactly the same way over one complete revolution.

Any practical solution of the blade motion problem requires one to consider only a finite number of blade degrees-of-freedom. In almost all cases, only one degree-of-freedom needs to be considered.

For the present development, suppose M degrees-of-freedom are chosen to represent the flexible rotor blade. Define the blade state column q as MX1 column vector assembled from the β_j coordinates as follows:

$$q = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_M \end{bmatrix} \tag{45}$$

The g column is MX1 vector composed of all of the g_j forcing functions, assembled analogously to Eq. (45). With this notation, Eqs. (39) and (40) can be written in the compact form (dropping reference to the physical constants):

$$\ddot{q} = g\,(q, \dot{q}, \psi) \tag{46}$$

Generally, this equation must be solved numerically because of the difficulties that arise when one attempts to expand g. Classically, certain assumptions are made concerning the blade's aerodynamic characteristics. The g column is expanded, and the q and $\dot{q}$ dependent terms are transposed to the left side of (46). Further assumptions allow a Fourier series approach to be applicable to the resulting linear differential equation in time varying coefficients, until a closed form solution for q (t) is reached.

This approach is not necessary when numerical techniques can be employed. The classical approach also becomes seriously restrictive when special nonlinear rotor phenomena are being studied.

A convenient state variable notation can be defined for the q and $\dot{q}$ columns. Define the 2MX1 column vector ζ:

$$\zeta = \begin{pmatrix} q \\ \dot{q} \end{pmatrix} \tag{47}$$

To determine the blade motion numerically, first estimate a value of ζ at $\psi = 0$, and denote this state vector as $\zeta_e(0)$. With this estimate, $\zeta(t)$ can be calculated by numerical solution of (46). Denote the value of ζ at $\psi = 2\pi$, using $\zeta_e(0)$, as $\zeta_o(2\pi)$. If $\zeta_e(0)$ was correct, then

$$\zeta_o(2\pi) = \zeta_e(0) \tag{48}$$

because the blade motion is periodic over $0 \leq \psi < 2\pi$. Of course, condition (48) will seldom occur from the initial estimate. To determine the correct initial condition, an iteration process can be used that is, very similar to the trim search iteration used for the entire vehicle.

After $\zeta_o(2\pi)$ has been computed, each element of ζ_e can be perturbed (one at a time) a small amount, and a new $\zeta(2\pi)$ column can be computed for each element perturbation. From these 2M calculations of $\zeta(2\pi)$, the matrix Z_{Z0} can be assembled such that

$$\Delta\zeta(2\pi) = Z_{Z0}\,\Delta\zeta(0) \tag{49}$$

Each column, i, of Z_{Z0} is the column $\zeta_i(2\pi) - \zeta_o(2\pi)$, where $\zeta_i(2\pi)$ is the state vector computed with the i'th element of $\zeta_e(0)$ perturbed.

Now say that the true value of $\zeta(2\pi)$ is given by

$$\zeta(2\pi) = \zeta_o(2\pi) + \Delta\zeta(2\pi) \tag{50}$$

and that the true value of $\zeta(0)$ is given by

$$\zeta(0) = \zeta_e(0) + \Delta\zeta(0) \tag{51}$$

Since the blade motion is periodic,

$$\zeta(0) = \zeta(2\pi) \tag{52}$$

Thus, combining Eqs. (50), (51) and (52),

$$\zeta_e(0) + \Delta\zeta(0) = \zeta_o(2\pi) + Z_{Z0}\,\Delta\zeta(0) \tag{53}$$

Solving for $\Delta\zeta(0)$, the true initial condition is given by

$$\zeta(0) = \zeta_e(0) + \Delta\zeta(0) = \zeta_e(0) + (1 - Z_{Z0})^{-1}\left[\zeta_o(2\pi) - \zeta_e(0)\right] \tag{54}$$

Eq. (54) would provide the exact initial condition, if (46) were linear in q, q. Many times, the equation is quite linear, but at extreme operating conditions (blade stall, compressibility drag rise, etc.), (46) may be quite nonlinear. Such nonlinearity will cause the array Z_{Z0} to be a function of $\zeta_e(0)$. For these

cases, the process outlined above for finding $\zeta(0)$ may have to be repeated several times, using the computed $\zeta(0)$ from Eq. (54) for the estimate before the next iteration cycle. Note that during the trim-search-iteration process, this blade iteration process takes place automatically.

After $\zeta(0)$ is known, one more integration can be performed over $0 \leq \psi < 2\pi$. This time, Eq. (44) will be solved along with the blade motion equation. The result of this final sweep will, of course, represent the desired functional computation, (1).

The function (1) will be required for two different kinds of calculations:

(a) The values of the variables in the argument of independent of any other set of values for these variables.

(b) The values of the variables in the argument of (1) are removed from those values used for a previous computation of f_1 by only an infinitesimally small amount.

In case (a), the matrix Z_{ZO} will have to be computed. In case (b), however, the Z_{ZO} computed for the initial solution for f_1 (using the unperturbed values of the argument variables in (1) can be used again for the perturbed computation. This procedure will save considerable computer time and allow the rotor loads to be computed separately from loads of other vehicle elements without compromising program efficiency (i.e., $\zeta(0)$ can be found efficiently in the rotor computational routine; otherwise, $\zeta(0)$ would have to be included with the other elements of the t column during the trim search phase).

During the trim search phase of computation, the column $\zeta(0)$ will have to be inspected for convergence to a trim solution, along with the t and v columns.

II. EQUATION SUMMARY FOR THE MODULAR STABILITY DERIVATIVE PROGRAM

II.1 INTRODUCTION

Part I describes the Modular Stability Derivative Program (MOSTAB) in general terms. In Part I, an aircraft is represented by functional relationships (as defined by Eqs. (2), (7), (11), (12) and (14)) and by geometric relationships (Eqs. (3), (4) and (5)). The purpose of this work is to define the specific functional and geometric models of an aircraft presently used in the MOSTAB-B program. These mathematical models will undoubtedly be revised and expanded as MOSTAB-B is used to study specific vehicles operating in specific flight regimes.* It is believed, however, that the mathematical expressions derived here are quite general, and are sufficiently flexible to allow most modifications to be made with ease. In their present form, the equations will apply accurately to a broad variety of V/STOL (and conventional aircraft) configurations operating over large regions of their individual flight regimes. Modification of the MOSTAB equations will probably occur when boundary regions (involving special aerodynamic effects) are studied, or when vehicle configurations with very specialized components are considered.

Part II is divided into sections, most of which relate directly to the equations of Part I. Section 2 deals with aircraft element force generation, and shows the development of the equations required to represent functional Eqs. (2) of Part I. MOSTAB-B uses five basic subroutines to generate the general force column, f. These are FORCE, BODY, LIFT, SWEEP and ROTOR. Section 2 is divided into subsections, each addressing one of these vehicle element subroutines. Because of the complexity of the aerodynamic rotor analysis, the basic rotor equations are derived in Part III. The general equations presented in this part are simplified and re-presented (as programmed in MOSTAB-B) in Section 2D.

Section 3 shows the derivation of equations for the geometric matrices L and G, as defined by expressions (3) and (4) of Part I.

Sections 4, 5, 6 and 7 show derivations of the equations used in MOSTAB to generate the functional relationships defined by Eqs. (7), (11), (12) and (14) (respectively) of Part I.

Section 8 discusses the general matrix operation subroutines used in MOSTAB-B. The equations used in the Eulerian coordinate system transformation subroutine (EULER) are presented. The other operational subroutines are discussed without equations, since such subroutines are widely used in computer applications and require no definition specifically constructed for MOSTAB-B.

---

*The -B code number on the title MOSTAB-B denotes the B version of the program. Versions including advanced aerodynamics additional element models (e.g., turbojets), will be given different version codes.

Section 9 shows the derivation of the transformations required to transform stability derivative matrices expressed in overall vehicle coordinates to center-of-gravity and stability axis system coordinates. No reference to this section, or to Section 8, is made in the general MOSTAB-B program description in Chapter V of the main text or in Part I.

II.2 AERODYNAMIC ELEMENT FORCE GENERATION

A. Subroutine FORCE

In Part I it was assumed that a force column, f, could be calculated knowing state vector columns, v_I, v_A, $\dot{v}_I$ and c (Eq. 2 in Reference 1, or Eq. 1 of Appendix I, denoted here as Eq.1F).

$$f = f(v_I,\ v_A,\ \dot{v}_I,\ c,\ K_j,\ j = 1,2 \cdots) \qquad (1F)$$

where K_j are physical constants pertaining to a specific aircraft.

A subroutine named FORCE performs the functional operation defined by Eq. (1F). Given the columns shown as arguments in Eq. (1F), FORCE returns vehicle element loadings.

Two options are available in FORCE. Option 1 tells FORCE to determine all new elements for f. Option 2 tells FORCE to compute only six elements of f. The particular six-element subcolumn to be generated is specified when FORCE is called, and represents the load generated by a single aerodynamic element.

At the present time, FORCE calls three subroutines for the purpose of calculating vehicle element aerodynamic loading. These are BODY, LIFT and ROTOR. BODY computes the aerodynamic loads generated by fuselages, nacelles, etc. LIFT produces load values for nonrotating lifting vehicle elements such as wings and empennage surfaces. ROTOR, in conjunction with a subroutine called SWEEP, calculates loadings produced by helicopter main and tail rotors, propellers, etc.

In the future, other forcing element routines can be added to this library of three to represent such additional components as turbojet engines and rockets.

FORCE contains no aerodynamic expressions, but is a logical subroutine which directs the operation of calculating aerodynamic forces. FORCE determines which aerodynamic elements are to be exercised, addresses the proper load-calculating subroutines (BODY, LIFT or ROTOR), and assigns the proper set of physical constants to a COMMON region before addressing the load-computing routine(s). The computed loads are transferred back to the main program through FORCE.

Figure 1 shows the general operation of the subroutines FORCE, BODY, LIFT, ROTOR and SWEEP as they interlace to compute aerodynamic element loads. BODY, LIFT and ROTOR use the Euler resolution subroutine EULER. This general use of EULER is not shown on the diagram. The ROTOR-SWEEP interface will be outlined in the ROTOR section of this report.

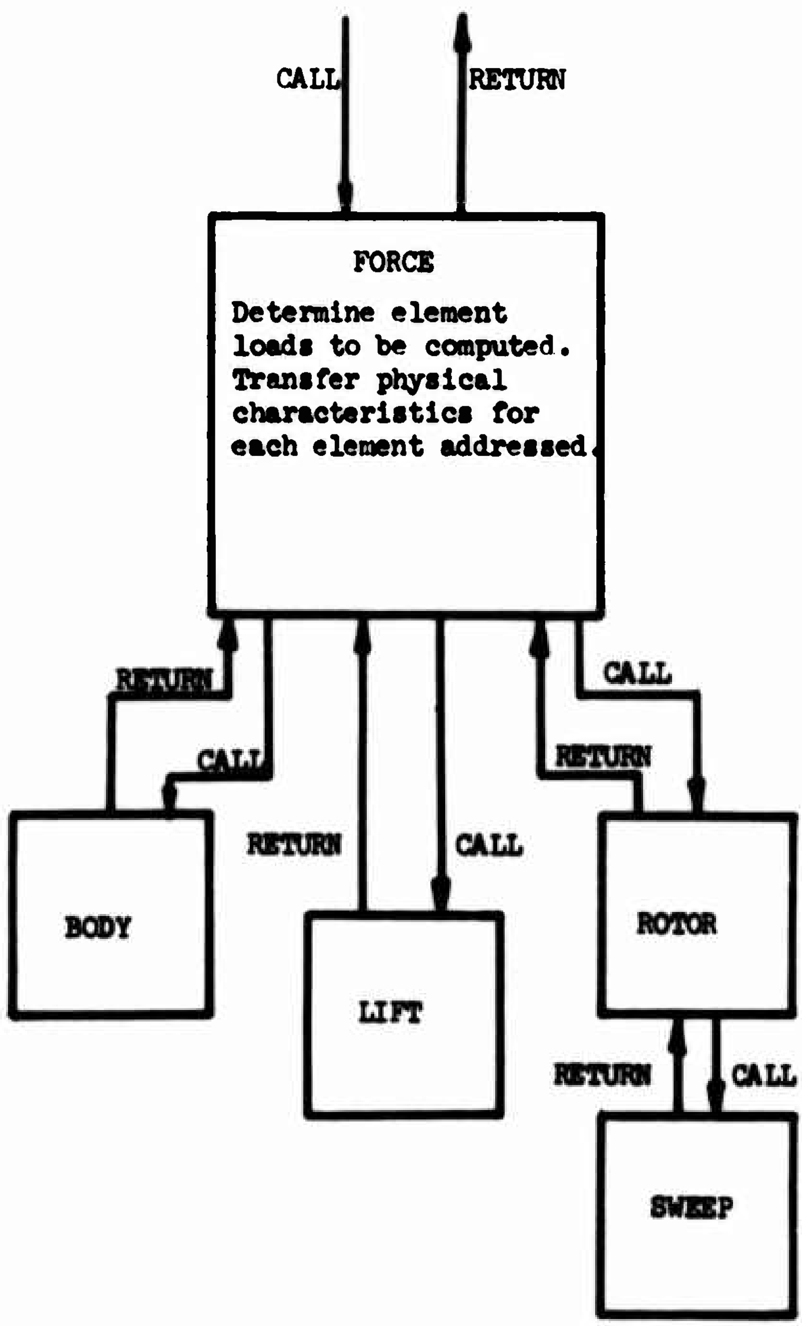

Figure 1. General Operation of Subroutines FORCE, BODY, LIFT, ROTOR, and SWEEP.

B. Subroutine BODY

This subroutine receives the following information when it is called:

(a) The three translational and three rotational airspeed components at the body reference point. These velocity components are given in overall vehicle axes (see Reference 1).

(b) Three Euler angles ψ_b, θ_b, φ_b which are used to rotate vectors expressed in overall vehicle coordinates to a reference system conveniently related to the body. The reverse resolution is also done with these angles.

(c) The aerodynamic coefficients, characteristic areas, lengths, etc., representing the characteristics of the specific aerodynamic body being considered.

At the present time, BODY incorporates equations which derive from the following six aerodynamic coefficient expressions

$$C_x = -\left(C_o + C_1\,\alpha + C_2\,\beta\right) \tag{55}$$

$$C_y = -\left(C_{yo} + C_{y1}\,\beta\right) \tag{56}$$

$$C_z = -\left(C_{zo} + C_{z1}\,\alpha\right) \tag{57}$$

$$C_l = 0 \tag{58}$$

$$C_m = C_{mo} + C_{m1}\,\alpha \tag{59}$$

$$C_n = C_{no} + C_{n1}\,\beta \tag{60}$$

The coefficients in the above equations are functions of the specific configuration of the aerodynamic body, and are input to the MOSTAB program. At the present time, these coefficients are assumed to be constants. A more sophisticated set of expressions would allow these coefficients to vary with local aerodynamic conditions.

Dimensional forces and moments are computed, using the coefficient expressions (55 - 60) and relying on the following assumptions:

(a) The dynamic pressure at the body is given by

$$q_a = 1/2\,\rho\,u_b^2 \tag{61}$$

b) The longitudinal angle of attack, α, is sufficiently small so that

$$\alpha \cong \frac{w_b}{u_b} \tag{62}$$

c) The sideslip angle, β, is sufficiently small so that

$$\beta \cong \frac{v_b}{u_b} \tag{63}$$

Eqs. (1B - 9B) can be combined to yield the following dimensional force/moment equations in coordinates of the aerodynamic body being considered:

$$X_b = -\ 1/2\ \rho A_b \left[u_b^{\ 2}\ C_o + u_b w_b\ C_1 + u_b v_b\ C_2\right] \tag{64}$$

$$Y_b = -\ 1/2\ \rho A_b \left[u_b^{\ 2}\ C_{yo} + u_b\ v_b\ C_{y1}\right] \tag{65}$$

$$Z_b = -\ 1/2\ \rho A_b \left[u_b^{\ 2}\ C_{zo} + u_b\ w_b\ C_{z1}\right] \tag{66}$$

$$L_b = 0 \tag{67}$$

$$M_b = 1/2\ \rho A_b \ell_b \left[u_b^{\ 2}\ C_{mo} + u_b\ w_n\ C_{m1}\right] \tag{68}$$

$$N_b = 1/2\ \rho A_b \ell_b \left[u_b^{\ 2}\ C_{no} + u_b\ v_b\ C_{n1}\right] \tag{69}$$

Eqs. (64) - (69) are those presently programmed in BODY. The Euler angles ψ_b, θ_b, φ_b are used with a standard Euler angle subroutine called EULER to derive local airspeed components u_b, v_b, w_b, p_b, q_b, r_b from the given vector V_A. Eqs. (64) - (69) are executed, and EULER is called again, this time to rotate force and moment elements X_b, Y_b, Z_b, L_b, M_b, N_b back to overall vehicle axes x,y,z. Of course, EULER uses the given angles in the order $-\varphi_b$, $-\theta_b$, $-\psi_b$ to perform the force and moment resolution back to vehicle coordinates.

If MOSTAB is being used to study an area of the flight envelope where variable coefficients in Eqs. (64) - (69) are important, then suitable variable coefficient expressions can be formed such that Eqs. (64) - (69) still hold. The modular construction of MOSTAB makes it easy to expand these equations to include nonconstant coefficients without interrupting the overall program function. Such changes in the aerodynamic expressions occur locally in the BODY subroutine, and have no influence on other parts of the program.

C. Subroutine LIFT

This subroutine calculates loads produced by a nonrotating lifting surface (LS). The equations are formulated in a manner very similar to that used to assemble the equations for BODY. When LIFT is called, the following information is available:

(a) The three translational and three rotational airspeed components of the LS at its reference point. These components refer to overall vehicle coordinate axes.

(b) Three Euler angles ψ_w, θ_w, φ_w, which are used to rotate vectors expressed in vehicle coordinates a reference system conveniently related to the LS (hereinafter called LS coordinates). The reverse resolution is also done with these angles.

(c) The aerodynamic coefficients, characteristic areas, lengths, etc., representing the characteristics of the specific LS being considered.

Two sets of lifting surface models are presently incorporated in the LIFT subroutine: (1) models appropriate for angles-of-attack, α, bounded between values of $\pm$ .2 radian ($\approx 12^{\circ}$); and (2) models for angles-of-attack greater than .2 radian. These two aerodynamic models are discussed below under separate headings.

(1) Aerodynamic Lifting Surface Models, $-.2 \leq \alpha \leq .2$ Radian

The models presented below are appropriate for most flight conditions, because lifting surfaces operate below stall during the normal operation of most flight vehicles. Some vehicles (e.g., helicopters) do fly under conditions where lifting surfaces are stalled, but most of these cases involve such low dynamic pressures that the surfaces have negligible effect on the flight characteristics.

The following basic aerodynamic expressions are incorporated in the MOSTAB-B LIFT subroutine:

$$C_D = + \left[C_{DO} + C_{D1} \left(\alpha - \alpha_{WCLD}\right) + C_{D2} \left(\alpha - \alpha_{WCLD}\right)^2 \right] \qquad (70)$$

$$C_y = 0 \qquad (71)$$

$$C_L = + a_w \, \alpha \qquad (72)$$

$$C_l = - a_w \, \Gamma \left[\frac{1 + 2\lambda_w}{6(1 + \lambda_w)}\right] \beta - \frac{a_w \, b_w}{u_w} \left[\frac{1 + 3\lambda_w}{24(1 + \lambda_w)}\right] p_w$$

$$+ \frac{b_w \, C_L}{u_w} \left[\frac{1 + 3\lambda_w}{12(1 + \lambda_w)}\right] r_w \qquad (73)$$

$$C_m = + C_{mo} + C_{m\alpha} \, \alpha \qquad (74)$$

$$C_n = - \frac{b_w}{u_w} \left[\frac{1 + 3\lambda_w}{24(1 + \lambda_w)}\right] \left(C_L - C_{D1}\right) p_w \qquad (75)$$

Eqs. (70) - (75) are given in most standard texts (e.g., Ref. 1) and apply particularly to unswept subsonic wings. All of these expressions are derived in Reference 2.* These wing coefficient expressions are very elementary equations and certainly will be revised and expanded in the future. Expressions for ailerons, wing sweep, stall and even compressibility effects can easily be added to this basic set of equations. Many other special wing effects will be accounted for by suitable interference velocity models mechanized in subroutine WASH.**

Note that lift and drag coefficients are used in (70) and (72), instead of the local IS axis system coefficients C_x and C_z. This has been done because (70) and (72) are the most familiar force expressions for a wing. C_x and C_z are easily derived from C_L and C_D by referring to Figure 2.

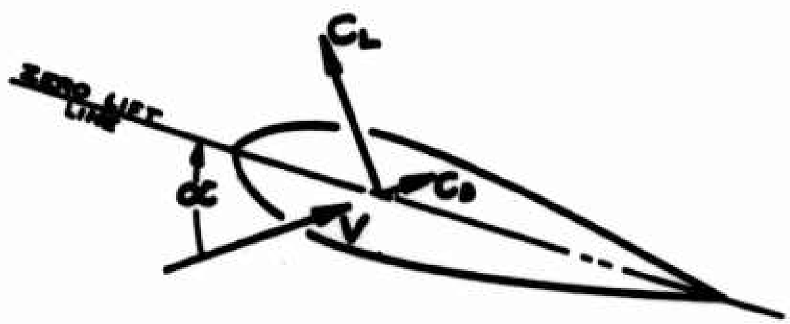

Figure 2. Basic Lift and Drag Coefficient Relationships.

---

* The dihedral angle, Γ, is an "effective" dihedral angle. Its numerical value depends on the wing geometric dihedral angle, and its position on the element to which it is attached. See Reference 9, for example, which shows C_{L_β} corrections which can be made to account for vertical position of a wing on a fuselage.

** For example, aileron deflection causes yawing moments to be developed by the vertical fin on a conventional airplane. This coupling is caused by rotational interference velocity components, originating because of a wing rolling moment, and eventually impinging on the vertical empennage surface. This effect is easily included in MOSTAB by assembling the proper elements for the characteristic area matrix and the interference velocity coupling matrix (see the Subroutine WASH section of this part for a discussion of these matrices).

Choose ψ_w, θ_w, φ_w so that the LS x axis lies parallel to the overall wing line of zero lift. Then

$$C_x = C_L \sin \alpha - C_D \cos \alpha \tag{76}$$

$$C_z = - C_L \cos \alpha - C_D \sin \alpha \tag{77}$$

from inspection of Figure 2.

Eqs. (70) - (75) and the derived Eqs. (76) and (77) require that ψ_w, θ_w, φ_w be chosen so that

(1) The x and z axes of the LS axis system lie in the wing's plane of symmetry.

(2) The x axis of the LS system is parallel to the wing's overall line of zero lift.

For sufficiently small α and β, the following equations can be written:

$$\sin \alpha \cong \tan \alpha \cong \frac{w_w}{u_w} , \quad \cos \alpha = 1 \tag{78}$$

$$\beta \cong \frac{v_w}{u_w} \tag{79}$$

$$q_a \cong 1/2 \, \rho u_w^{\ 2} \tag{80}$$

With these assumptions, Eqs. (70) - (77) can be combined and dimensionalized to yield the dimensional wing force/moment equations:

$$X_W = -1/2\ \rho S_W \left[-a_W w_W^2 + C_{Do} u_W^2 + C_{D1}\ u_W \left(w_W - \alpha_{WCLD} u_W \right) + C_{D2} \left(w_W - \alpha_{WCLD}\ u_W \right)^2 \right] \tag{81}$$

$$Y_W = 0 \tag{82}$$

$$Z_W = -\ 1/2\ \rho S_W \left[\left(a_W + C_{DO} \right) u_W\ w_W + C_{D1}\ w_W \left(w_W - \alpha_{WCLD} u_W \right) \right] \tag{83}$$

$$L_W = -\ 1/2\ \rho S_W\ b_W \left\{ a_W\ \Gamma \left[\frac{1 + 2\lambda_W}{6(1 + \lambda_W)} \right] v_W\ u_W + a_W\ b_W \left[\frac{1 + 3\ \lambda_W}{24\ (1 + \lambda_W)} \right] \left(p_W\ u_W - r_W\ w_W \right) \right\} \tag{84}$$

$$M_W = +\ 1/2\ \rho S_W\ \bar{c}\ \left[C_{mo}\ u_W^2 + C_{m\alpha}\ u_W\ w_W \right] \tag{85}$$

$$N_W = -\ 1/2\ \rho S_W b_W \left\{ b_W \left[\frac{1 + 3\ \lambda_W}{24(1 + \lambda_W)} \right] \left(a_W\ w_W - C_{D_1}\ u_W \right) p_W \right\} \tag{86}$$

Eqs. (81) - (86) are executed in LIFT in much the same way loads are calculated in the BODY subroutine. The air-speed components in v_A applicable to the LS are rotated through angles ψ_w, θ_w, φ_w, and become u_w, v_w, w_w, p_w, q_w, r_w. Eqs. (12L)-(17L) are executed, and the resulting load components are rotated back to overall vehicle coordinates, through angles $-\varphi_w$, $-\theta_w$, $-\psi_w$. The Euler rotations are performed by the general subroutine, EULER.

If flight regions are studied wherein the constant coefficient approximation is invalid, then Eqs. (81) - (86) can still be used, but with suitable expressions for the variable coefficients. Constant coefficients assumptions on Eqs. (81) - (86) are valid over a large portion of most V/STOL flight envelopes.

It is important to note that the classical expressions for wing three-dimensional effects are absent from the equations (i.e., induced drag is not included explicitly in Eqs. (81) through (86).* The influence of wing downwash on wing loads comes into the equations, because wing interference velocities are included in the vector v_A when LIFT is called. The expressions that deal with interference velocities at the LS caused by the same LS are included in the interference velocity subroutine WASH.

The wing equations may be applied to empennage surfaces which do not have a plane-of-symmetry (e.g., a standard vertical stabilizer). Airfoils with camber, or special offset angles and positions can be considered also. The reference point and Euler angles can be chosen, along with suitable aerodynamic coefficients, such that Eqs. (81) - (86) are reasonably accurate for such a surface. The reference to "wing" equations and "lifting surface" equations is intended to be synonymous here.

* The coefficients in Eq. (70) are the factors of the parabolic profile drag polar. Induced effects do not influence these coefficients.

Of course, a wing producing 2900 pounds of **thrust** (a very sizable number for a 200 ft^2 wing) is a very unusual wing! The equations have seriously broken down, having a profound influence on the performance prediction under the example conditions.

To avoid such serious breakdown, MOSTAB-B bypasses Eqs. (81) - (86) when $|\alpha|$ is larger than .2 radian and calculates lifting surface loads defined by the models which are presented below. Only lift and drag are considered, so Y_w, L_w, M_w and N_w (notation of Eqs. (81) - (86) are set to zero. Although this large angle model is very rough, it suffices for most cases because, the model is seldom required except when dynamic pressures are low.

Figures 4 and 5, taken from Reference 10, were used as a basis for modeling the large angle lift and drag coefficient functions. Figures 6 and 7 depict the actual models presently mechanized in LIFT for large angles of attack.

The large angle C_D curve (Figure 6) is a parabola with a maximum value of 1.5. Note that maximum C_D values on Figure 4 vary from 1.7 to 2.35. These higher $C_D|_{max}$ values are not realistic for a practical lifting surface, because they are essentially two-dimensional results.

(2) **Aerodynamic Lifting Surface Models; $.2 \geq \alpha \geq -.2$.**

The LS models presented in paragraph (1) above are appropriate for most aircraft flight conditions. Even if the LS is stalled, it usually has little effect on flight dynamics, because such stalling almost always occurs at low dynamic pressures.

When $|\alpha|$ gets larger than approximately 12°, even if the dynamic pressure is low, Eqs. (81) - (86) break down in such a way to produce large numbers (which, of course, are in error). For example consider Figure 3, which shows a wing on a hovering compound helicopter immersed in a rotor wash of 45 ft/sec. (a typical hovering rotor downwash velocity at a point somewhat downstream from the rotor. Figure 3 also shows the wing's x_w - z_w axes, and the X_w, Z_w forces. If Eqs. (81) and (83) are used, values of X_w and Z_w are calculated as follows (for convenience, use C_{D_0}, and $C_{D_2} = 0$, $\rho = .002377$, $a_w = 6.0$):

$$X_w = 2900 \sim \text{lb}$$

$$Z_w \cong 0.0$$

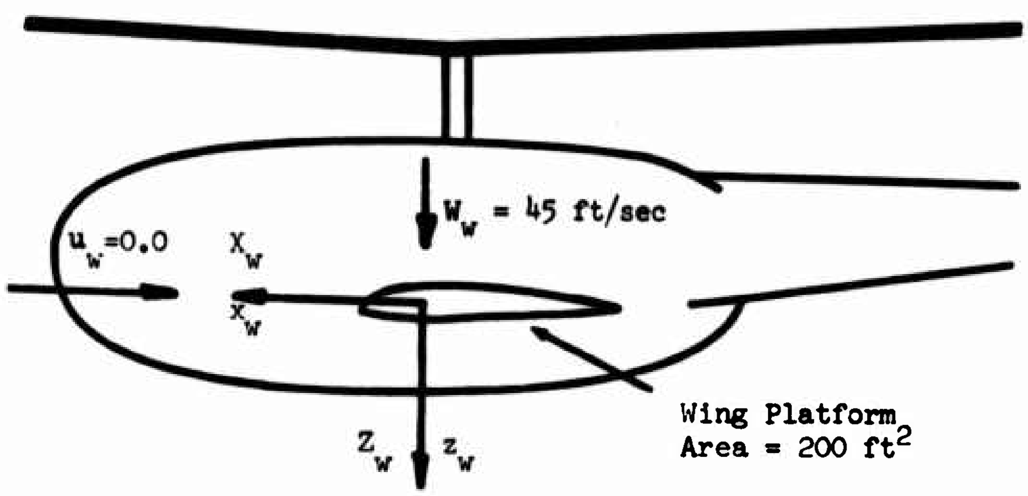

Figure 3. Rotor-Wing Interaction.

Of course, a wing producing 2900 pounds of thrust (a very sizable number for a 200 ft^2 wing) is a very unusual wing! The equations have seriously broken down, having a profound influence on the performance prediction under the example conditions.

To avoid such serious breakdown, MOSTAB-B bypasses Eqs. (81)-(86) when $|\alpha|$ is larger than .2 radians, and calculates lifting surface loads defined by the models which are presented below. Only lift and drag are considered, so Y_w, L_w, M_w and N_w (notation of Eqs. (81) - (86) are set to zero. Although this large angle model is very rough, it suffices for most cases because, as mentioned before, the model is seldom required except when dynamic pressures are low.

Figures 4 and 5, taken from Reference 10, were used as a basis for modeling the large angle lift and drag coefficient functions. Figures 6 and 7 depict the actual models presently mechanized in LIFT for large angles of attack.

The large angle C_D curve (Figure 6) is a parabola with a maximum value of 1.5. Note that maximum C_D values on Figure 4 vary from 1.7 to 2.35. These higher $C_D|_{max}$ values are not realistic for a practical lifting surface, because they are essentially two-dimensional results.

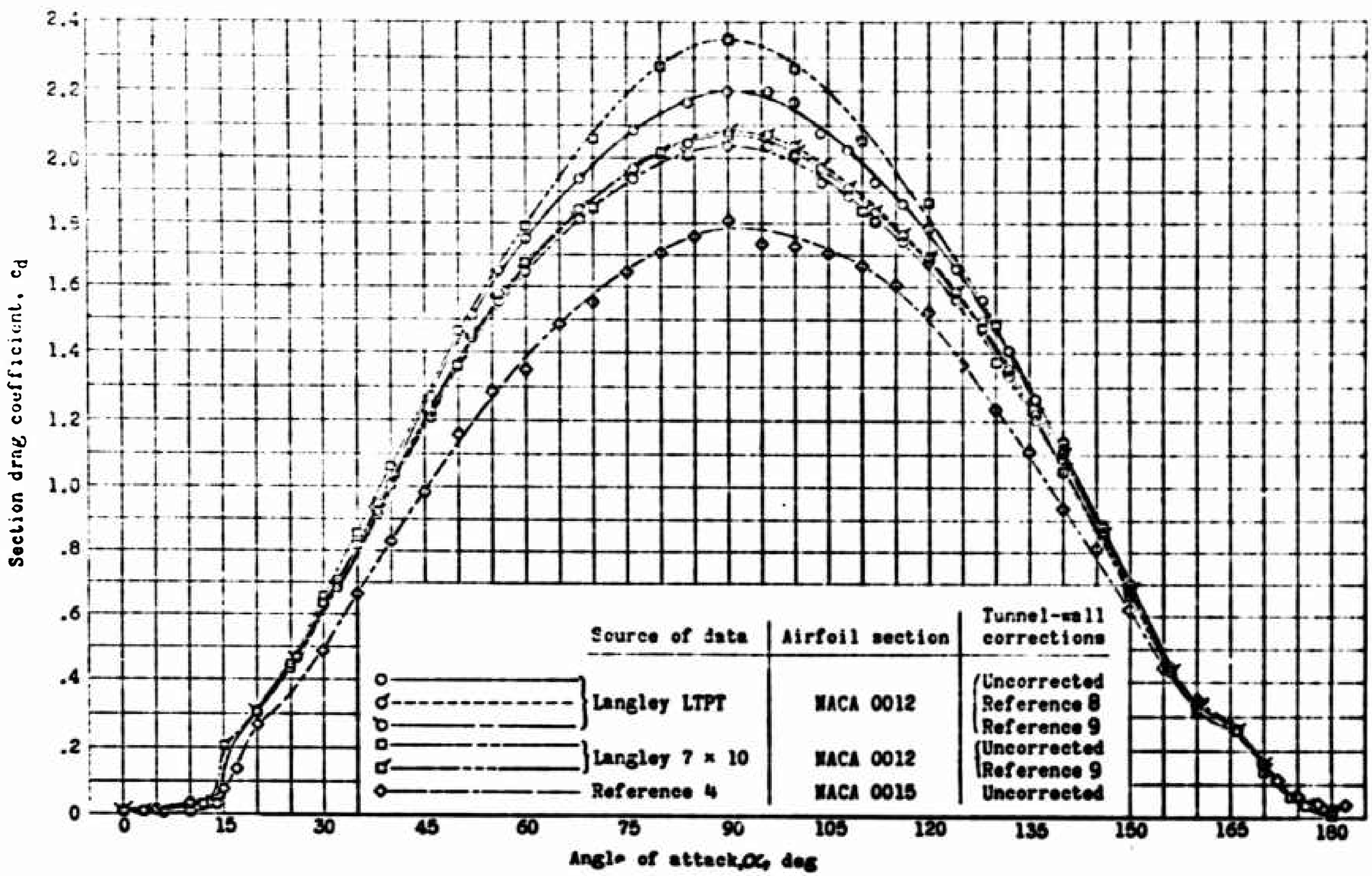

Figure 4. Large Angle C_D Curve From Reference 10.

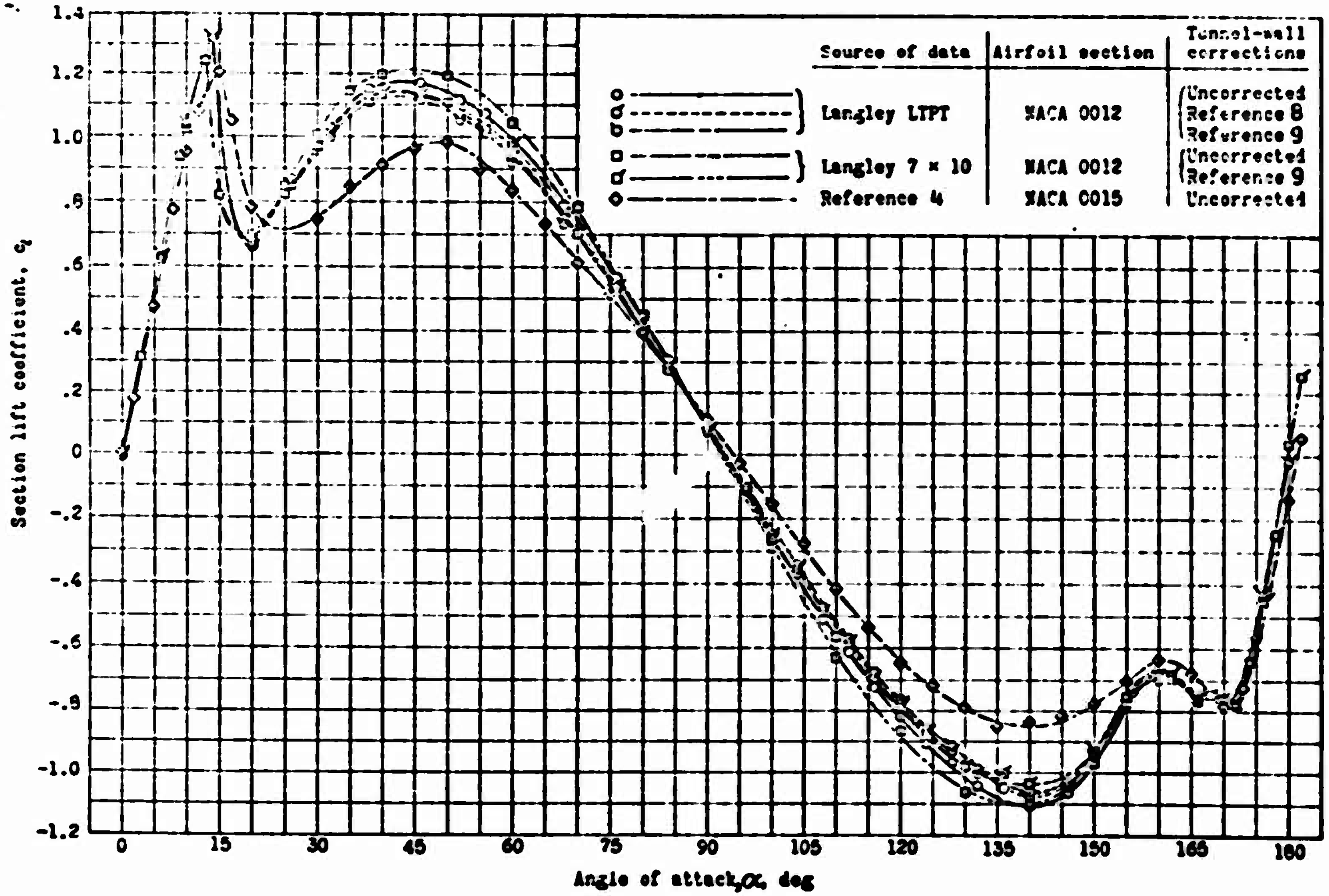

Figure 5. Large Angle C_L Curve From Reference 10.

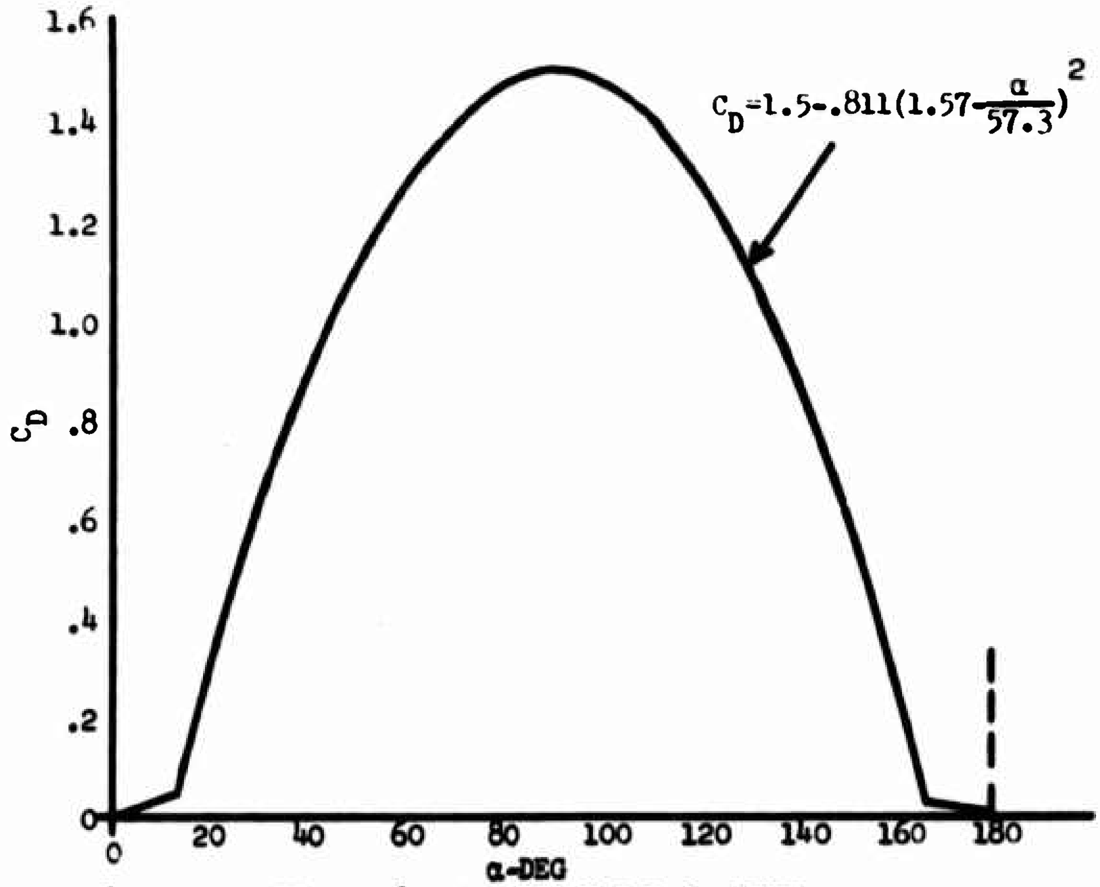

Figure 6. - C_D vs α Model in LIFT.

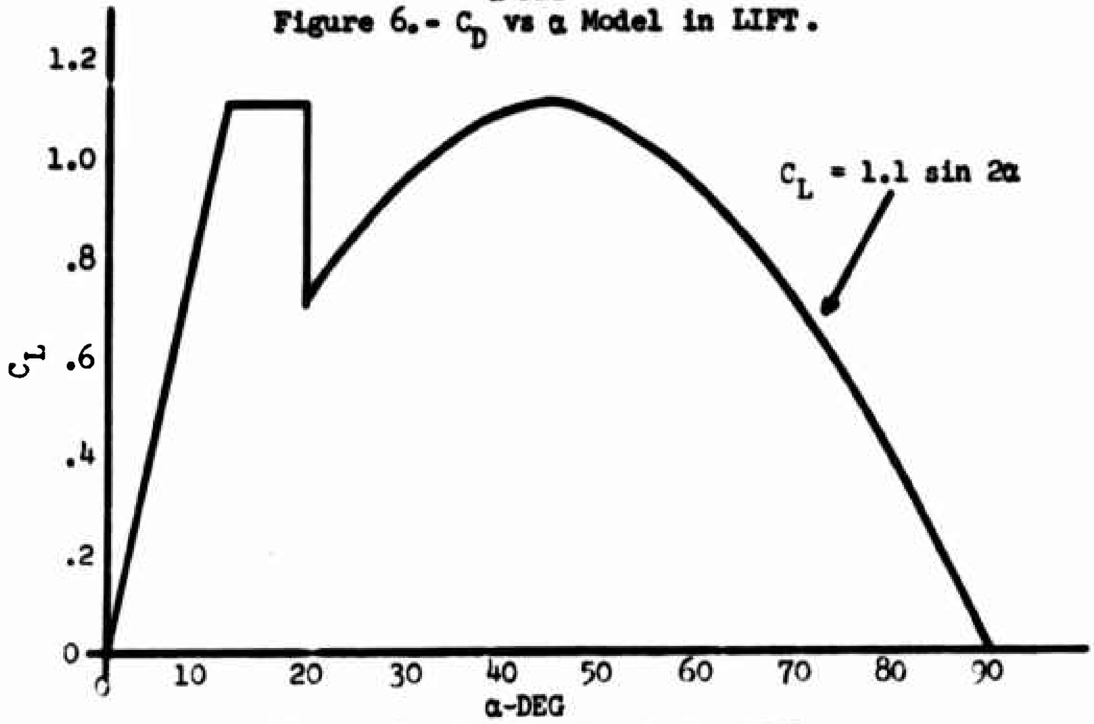

Figure 7. - C_L vs α Model in LIFT.

Figure 8 demonstrates this point by showing the flat plate drag coefficient for rectangular plates as a function of aspect ratio (taken from Reference 11). The C_{D_0} value of 2.0 for infinite aspect ratio is

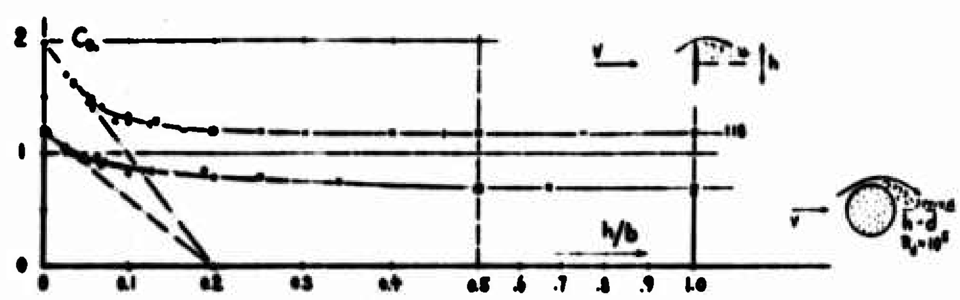

Figure 8. Drag Coefficients of Rectangular Plates and Circular Cylinders as a Function of Their Height (or Diameter) to Span Ratio.

compatible with the curves of Figure 4. Plates (i.e., rectangular wings at $\alpha = 90^{\circ}$) with finite aspect ratios have $C_{D_{0|max}}$ values between 1.18 and 2.0, however, so a representative value of 1.5 was arbitrarily chosen for the LIFT model depicted by Figure 6. Based on Figure 4, the following logic is incorporated in LIFT to enable generation of C_D for $-180^{\circ} < \alpha < +180^{\circ}$. First, define the three basic equations:

$$C_{D|STALL} \triangleq 1.5 - .811 \left(1.57 - \frac{|\alpha|}{57.3}\right)^2 \qquad (87)$$

$$C_D\Big|_{|\alpha|<20^\circ} \triangleq C_{D_0} + C_{D_2}\, \alpha^2 \qquad (88)$$

$$C_D\Big|_{|\alpha|>160^\circ} \triangleq 2.75 \left(C_{D_0} + C_{D_2}\, \alpha^2\right) \qquad (89)$$

The coefficients C_{D_0} and C_{D_2} in Eqs. (88) and (89) are the same values as the coefficients in Eq. (81). The factor 2.75 for α value in the region $160^\circ<|\alpha|\leq 180^\circ$ was obtained from unpublished data which indicate "reverse flow" profile drag values approximately equal to 2.75 times the "forward flow" values. This factor will vary from airfoil to airfoil, of course, but 2.75 is a representative value. Using Eqs. (87) through (89), the 180° drag model in LIFT is defined by Table I:

TABLE I. DRAG CHARACTERISTICS

α Region	C_D
$12^\circ<\lvert\alpha\rvert\leq 20^\circ$	C_D = larger between $C_{D\|STALL}$ and $C_D\Big\|_{\lvert\alpha\rvert<20^\circ}$
$20^\circ<\lvert\alpha\rvert<160^\circ$	$C_D = C_{D\|STALL}$
$160^\circ\leq\lvert\alpha\rvert\leq 180^\circ$	C_D = larger between $C_{D\|STALL}$ and $C_D\Big\|_{\lvert\alpha\rvert>160^\circ}$

The lift coefficient function of Figure 7 represents a relatively good fit of the data of Figure 5 for $0 \leq \alpha \leq 90^\circ$. Table I defines the logic for determining C_L for $-180^\circ \leq \alpha \leq 180^\circ$, where a_w is the same lift curve slope used in Eq. (72).

TABLE II. LIFT CHARACTERISTICS

α Region	C_L
$12^\circ \leq \alpha \leq 20^\circ$	C_L = lesser between $a_w \left(\frac{\alpha}{57.3}\right)$ and 1.1
$20^\circ < \alpha < 160^\circ$	$C_L = 1.1 \sin 2\alpha$
$160^\circ \leq \alpha \leq 180^\circ$	C_L = greater between $a_w \left(\frac{\alpha - 180}{57.3}\right)$ and -1.1
$-20^\circ \leq \alpha \leq -12^\circ$	C_L = greater between + $a_w \left(\frac{\alpha}{57.3}\right)$ and -1.1
$-160^\circ < \alpha < -20^\circ$	$C_L = 1.1 \sin 2\alpha$
$-180^\circ \leq \alpha \leq -160^\circ$	C_L lesser between $a_w \left(\frac{\alpha + 180}{57.3}\right)$ and 1.1

After the logic of Tables I and II are executed to calculate C_L and C_D, Eqs. (76) and (77) are used in LIFT to calculate C_x and C_z (with no small angle assumptions on α), so that

$$X_w = -\frac{1}{2}\rho V^2 S C_x \tag{90}$$

and

$$Z_w = -\frac{1}{2}\rho V^2 S C_z \tag{91}$$

where

$$V^2 = U_w{}^2 + W_w{}^2 \tag{92}$$

As mentioned previously, $Y_w = L_w = N_w = M_w = 0.0$ for large α's in the present LIFT subroutine.

D. Subroutine SWEEP

SWEEP contains the rotor blade equations. These equations are integrated radially and azimuthally in SWEEP, a process which essentially "sweeps" the rotor disc to obtain loads and blade motions. At the present time, SWEEP is exclusively called by ROTOR. SWEEP has been designed for speed, because it is this routine which will absorb the most computer time during any MOSTAB run. Thus, certain time-consuming operations (subscripted variable usage, general coordinate transformations when some elements of the transformation matrix are zero, etc.) have been avoided whenever possible, particularly in the radial integration loop of the subroutine.

The subroutine receives the following information when it is called:

(a) The three translational and three rotational airspeed components at the rotor reference point, already resolved (by ROTOR) to hub axes.

(b) The three translational and three rotational inertial velocity components of the rotor reference point, resolved to hub axes.

(c) The time derivatives of the quantities given in (b).

(d) An index defining the options to be exercised in SWEEP.

(e) The rotor control settings (collective and cyclic pitch angles).

(f) The flapping angle and flapping velocity of the blade at ψ=0, if the rotor under consideration is a flexible bladed rotor. (Only two state variables for the blade are specified here, because MOSTAB presently uses one degree-of-freedom for each blade. If additional blade degrees-of-freedom are added, additional state variable pairs will be specified here, at ψ=0).

(g) The physical characteristics of the rotor, including certain constants computed in ROTOR prior to the calling of SWEEP.

(h) The number of radial and azimuthal elements to be used in the integrations.

An option index directs SWEEP to adhere to one of the following computational schedules:

(1) Compute the blade motion of a flexible blade, but do not compute shaft loads.

(2) Compute blade motion and shaft loads.

(3) Blade motion suppressed. Compute shaft loads only.

The work that follows deals primarily with option 2, which is the most general option. Options 1 and 3 are simply suppressed versions of 2.

1. Blade Analysis

Relatively general equations for rotor blade motion and loading are derived in **Part III. The axis systems, reference** lines, etc, are discussed in detail in **Part III. In this** section, the equations derived are simplified to the form presently incorporated in MOSTAB. The simplifications are predicated on the following restrictions and assumptions:*

(a) Rotor speed is constant ($\dot{\Omega} = 0$).

(b) Centrifugal force is the only inplane inertial force important to vehicle dynamics. Other inplane inertial forces are neglected. Of course, aerodynamic inplane forces are important, and these are included in MOSTAB.

(c) Flexible rotors have one degree-of-freedom: the first flapping mode.

(d) Elastic torsional deformation is not important.

The blade reference line (BRL) is defined in **Part III.** Because of assumption (c), the following expressions can be written for the coordinates of the blade reference line:

$$x(s,t) = - s \tag{93}$$

$$\dot{x}(s,t) = \ddot{x}\,(s,t) = 0 \tag{94}$$

$$y(s,t) = \dot{y}(s,t) = \ddot{y}\,(s,t) = 0 \tag{95}$$

$$z(s,t) = z_o(s) + z_1(s)\beta(t) \tag{96}$$

In Eq. (96), $z_1(s)$ corresponds to the eigenfunction $\Phi_1(x)$ (discussed in **Part III for the first blade flapping mode.** $\beta(t)$ is the generalized coordinate ($\eta_1(t)$ in **Part III) for** this mode. $z_o(s)$ is an "initial shape" function, which can be added to the normal mode analysis with no loss of generality. $z_o(s)$ is the shape of the blade when it is not vibrating, and when the generalized forcing function is zero.

---

* The modular nature of MOSTAB permits these restrictions to be relaxed, if necessary, by simple changes in the subroutines without having to revise the entire program.

The eigenfunction can be normalized (by simply changing the scaling on its generalized coordinate) in any desired manner. If $z_1(s)$ is normalized such that

$$z'(R) \triangleq \left. \frac{dz(s)}{ds} \right|_{s=R} = 1 \quad , \tag{97}$$

then β is the slope of the rotor blade at the tip (excluding the initial slope that may be contributed by $z_o'(R)$). Thus, β becomes the dynamic flapping angle of the blade used in classical helicopter analyses. Note that, for β positive with "upward flapping", the functions $z_o(s)$ and $z(s)$ are generally negative.

The blade motion equation, in terms of β, is

$$\ddot{\beta} + \omega^2 \beta = \frac{F_g}{M_g} \tag{98}$$

where ω is frequency of the first flapping mode of the blade. M_g is the generalized mass of the first flapping mode, and is given by

$$M_g = \int_o^R m(s) z_1^2(s) ds \tag{99}$$

F_g, of course, is the generalized forcing function, and is given by

$$F_g = \int_o^R z_1(s)\ f_z\ (s,t)\ ds \tag{100}$$

The function $f(s,t)$ is the external BRL loading function less the acceleration terms used in the vibration analysis to get ω and z_1.

$$f_z(s,t) = p_z(s,t) + m(s)\ddot{z} \tag{101}$$

where $p_z(s,t)$ is the total distributed loading function on the BRL due to inertial "apparent" forces and aerodynamic forces.

$$p_z(s,t) = p_{zi}(s,t) + p_{za}(s,t) \tag{102}$$

The simplified inertial distributed loading functions presently incorporated in MOSTAB are

$$p_{xi}(s,t) = - m(s)\Omega^2 s \tag{103}$$

$$p_{yi}(s,t) = 0 \tag{104}$$

$$p_{zi}(s,t) = - m(s) \left\{ g_z + \ddot{z} + s\left[(\dot{p} - 2\Omega q)\sin\psi + (\dot{q} + 2\Omega p)\cos\psi\right]\right\} \tag{105}$$

These equations derive directly from Eqs. (293), (294) and (295) in **Part III. The simple forms of (103) and (104) are** attributable to assumption (b) of **this section.** Eq. (105) has been simplified to a greater degree than allowed by the assumptions (a)-(d) or by Eqs. (93) - (96), in that quadratic terms in p,q and r have been neglected.

The airspeed at a blade element is given by Eq. **(302)**. With present MOSTAB assumptions, this airspeed is (in rotor coordinates) is

$$\bar{V}_A \triangleq \frac{d^* \bar{h}}{dt} = \grave{i}\left[u_A \cos\psi - v_A \sin\psi + z(p_A \sin\psi + q_A \cos\psi)\right]$$

$$+ j\left[u_A \sin\psi + v_A \cos\psi + s(\Omega - r_A) - z(p_A \cos\psi - q_A \sin\psi)\right]$$

$$+ \grave{k}\left[w_A + \dot{z} + s(p_A \sin\psi + q_A \cos\psi)\right] \tag{106}$$

Eq. (305), Part III is used to resolve this airspeed expression to "blade cooruinates", as follows.

$$\begin{pmatrix} u_s \\ u_c \\ u_n \end{pmatrix} = T\,\overline{V}_A \tag{107}$$

where u_s is the spanwise airflow u_c is the chordwise airflow, and u_n is the normal-to-chord airflow at a blade section located at s.

The transformation matrix, T, can be simplified to the following form:

$$T = \begin{bmatrix} 1 & 0 & -z' \\ -\theta z' & 1-\frac{\theta^2}{2} & -\theta \\ z' & \theta & 1-\frac{\theta^2}{2} \end{bmatrix} \tag{108}$$

where ζ has been set to $-\theta$, and trigonometric approximations

$$\sin\theta = \theta$$
$$\cos\theta = 1 - \frac{1}{2}\theta^2$$

have been incorporated. Note that cubic and higher order products in small angles z' and θ have also been discarded from the T matrix approximated by Eq. (108), and quadratic terms in z' have been neglected.

Since no elastic torsional motion is included, the blade angle, θ, is given only linear twist:

$$\theta = \theta_o - A_{1s}\cos\psi - B_{1s}\sin\psi + \theta_1\left(\frac{s}{R}\right) + \delta_3\beta \tag{109}$$

where θ_o is the collective pitch angle, A_{1s} and B_{1s} are the usual lateral and longitudinal cyclic pitch angles (angles between the rotor control plane and the shaft normal plane), θ_1 is the linear twist angle, and δ_3 is the geometric flap-pitch coupling coefficient.

Eqs. (315) and (316) give the simplified aerodynamic forcing functions presently used in MOSTAB.

$$f_n = -\frac{\rho c}{2}\left[(a + \delta_o)u_n u_c + \delta_1 u_n^2\right] \tag{110}$$

$$f_c = -\frac{\rho c}{2}\left[(\delta_2 - a)\, u_n^2 + \delta_o u_c^2 + \delta_1 u_n u_c\right] \tag{111}$$

Assuming no aerodynamic loading in the blade span direction, Eqs. (110) and (111) can be used with the inverse (i.e., the transpose for an Eulerian transformation matrix) of the T matrix to get the distributed aerodynamic loading functions in rotor coordinates. The results of this operation, in terms of f_n and f_c, are:

$$\begin{pmatrix} p_{xa} \\ p_{yz} \\ p_{za} \end{pmatrix} = T^T \begin{pmatrix} 0 \\ f_c \\ f_n \end{pmatrix} \tag{112}$$

Eqs. (103) - (105) and (112) can be summed directly to get the simplified BRL loading functions used in MOSTAB:

$$p_x(s,t) = p_{xi}(s,t) + p_{xa}(s,t) \tag{113}$$

$$p_y(s,t) = p_{ya}(s,t) \tag{114}$$

$$p_z(s,t) = p_{zi}(s,t) + p_{za}(s,t) \tag{115}$$

The shaft loading expressions for one blade are given in **Part III (Eqs. (337) - (342). It is desirable to separate each** of these load components into the inertial contribution and the aerodynamic contribution. This process saves computer time by preventing certain constant inertial integrations (which need to be computed only once) from being repeated. The same case is true for the generalized forcing function given by Eq. (100).

The generalized force integral (from Eq. (100)) and **the shaft loading integrals (from Part III) are written below.** They are split into inertial and aerodynamic contributions, and several terms are dropped due to simplifications embodied in Eqs. (93) - (96) and (103) - (105).

$$F_g = F_{ga} + F_{gi} = \int_0^R z_1\, p_{za}\, ds - \int_0^R z_1 m \left\{ g_z + s\left[\left(\dot{p} - 2\Omega q\right)\sin\psi \right.\right.$$

$$\left.\left. + (\dot{q} + 2\Omega p)\cos\psi \right] \right\} ds \tag{116}$$

$$X_r = X_{ra} + X_{ri} = \int_0^R p_{xa}\, ds - \int_0^R m\Omega^2 s\, ds \tag{117}$$

$$Y_r = Y_{ra} + Y_{ri} = \int_0^R p_{ya}\, ds \tag{118}$$

$$Z_r = Z_{ra} + Z_{ri} = \int_0^R p_{za}\, ds - \int_0^R m \left\{ g_z + \ddot{z} + s\left[\left(\dot{p} - 2\Omega q\right)\sin\psi \right.\right.$$

$$\left.\left. + (\dot{q} + 2\Omega p)\cos\psi \right] \right\} ds \tag{119}$$

$$L_r = L_{ra} + L_{ri} = - \int_0^R z\, p_{ya}\, ds \tag{120}$$

$$M_r = M_{ra} + M_{ri} = \int_0^R \left[z\, p_{xa} + s\, p_{za} \right] ds - \int_0^R z\, m\Omega^2 s\, ds$$

$$- \int_0^R ms \left\{ g_z + \ddot{z} + s\left[(\dot{p} - 2\Omega q)\sin\psi + (\dot{q} + 2\Omega p)\cos\psi \right] \right\} \tag{121}$$

$$N_r = N_{ra} + N_{ri} = - \int_0^R s\, p_{ya}\, ds \tag{122}$$

The second integral on the right side of Eq. (117) can be dropped, since its effect will ultimately cancel among all of the rotor blades.

Several terms in Eqs. (116) - (122) must be dropped, in order to make the equations consistent with each other. Several other terms can be omitted by looking forward to Eqs. (149) - (154). (Expression of rotor loads in nonrotating coordinates).

First consider the question of consistency. Inplane inertial forces were omitted from the analysis. This omission caused the contributions to X_r, Y_r and N_r from accelerations g_x, g_y and $\dot{r}$ to vanish. The rotor makes such inplane load contributions simply because it has mass. These effects can be included in the analysis of a flying vehicle by adding suitable terms to the nonrotating airframe mass and inertia tensor. To do this, simply include a circular lamina structure attached to the nonrotating airframe:

(a) The lamina plane is perpendicular to the rotor shaft and passes through the rotor reference point.

(b) The mass of the lamina equals the rotor mass.

(c) The polar moment of inertia of the lamina equals that of the rotor.

Since the added structure is a circular lamina, its diametrical moment of inertia equals half its polar inertia. The influence of the lamina will take the place of:

(1) The neglected g_x, g_y and $\dot{r}$ effects discussed above

(2) The $-\int_0^R mg_z \, ds$ term in Eq. (119).

(3) The $-\int_0^R ms^2 \dot{p} \sin \psi \, ds$ and $-\int_0^R ms^2 \dot{q} \cos \psi \, ds$ terms in Eq. (29).

Thus, to use the lamina substitution consistently, the terms listed in (2) and (3) above must be dropped from the equations.

The following terms can be dropped due to cancellation among the two or more blades in a complete rotor:

(a) The $-\int_0^R m\Omega^2 s \, ds$ term in Eq. (117).

(b) The terms containing sin ψ and cos ψ in Eq. (119).

(c) The $-\int_0^R ms \, g_z \, ds$ term in Eq. (121).

Although these inertial terms are justifiably dropped from the shaft loading equations, note that they are left intact in the generalized forcing Eq. (111). These terms can be left out of the shaft load equations (due to the lamina model or interblade cancellation), but since they do affect blade motion, they must be included in the blade generalized forcing function (Eq. 116).

The aerodynamic integrals in Eqs. (116) - (122) must be evaluated numerically, due to the complexity of the aerodynamic distributed loading functions. The inertial terms can be expressed in simple form by defining the following constant integrals:

$$I_1 = \int_0^R m z_1 ds \tag{123}$$

$$I_2 = \int_0^R m z_1 s \, ds \tag{124}$$

$$J_B = \int_0^R m s^2 ds \tag{125}$$

Note that Eq. (125) represents the second mass moment integral for the blade. In terms of the constant integrals defined by Eqs. (123) - (125), the inertial components in Eqs. (116) - (124) may be written (noting that $\ddot{z} = z\ddot{\beta}$ from Eq. (96):

$$F_{gi} = - I_1 \, g_z - I_2 \left[(\dot{p}-2\Omega q) \sin\psi + (\dot{q}+2\Omega p)\cos\psi\right] \tag{126}$$

$$X_{ri} = - 0 \tag{127}$$

$$Y_{ri} = 0 \tag{128}$$

$$Z_{ri} = - I_1 \ddot{\beta} \tag{129}$$

$$L_{ri} = 0 \tag{130}$$

$$M_{ri} = - I_2(\Omega^2\beta + \ddot{\beta}) - J_B \left[-2\Omega q \sin\psi + 2\Omega p \cos\psi \right] \qquad (131)$$

$$N_{ri} = 0 \qquad (132)$$

An important observation can be made from Eq. (131). In most helicopter applications,

$$\Omega^2\beta \cong - \ddot{\beta} \qquad (133)$$

This is exactly true if the natural blade frequency is Ω (as is the case for an articulated rotor with no flapping hinge offset) and the blade is vibrating at its natural frequency. The term $(\Omega^2\beta + \ddot{\beta})$ is seen to be a small difference between two very large quantities. This difference is the source of serious errors in many numerical determinations of rotor blade motion.

To avoid the numerical difficulty discussed above, Eq. (98) can be used to eliminate $\ddot{\beta}$. This can be done in Eq. (129) as well as in Eq. (131). The equations will be written with this substitution in the summary.

2. Equation Summary

The equations used in MOSTAB are summarized below. They are repeated from expressions given on the preceding pages. For convenience, they are renumbered as 1S through 27S.

$$z(s,t) = z_o(s) + z_1(s)\ \beta(t) \tag{1S}$$

$$\ddot{\beta} + \omega^2 \beta = \frac{F_g}{M_g} \tag{2S}$$

$$M_g = \int_o^R m\ z_1^{\ 2}\ ds \tag{3S}$$

$$p_{xi} = -m(s)\Omega^2 s \tag{4S}$$

$$p_{yi} = 0 \tag{5S}$$

$$p_{zi} = -\ m(s) \left\{ g_z + \ddot{z} + s \left[(\dot{p}-2\Omega q)\sin\psi + (\dot{q}+2\Omega p)\cos\psi \right] \right\} \tag{6S}$$

$$\begin{aligned} V_a = \dot{i} & \left[u_a \cos\psi - v_A \sin\psi + z\ (p_A \sin\psi + q_A \cos\psi) \right] \\ & + j \left[u_A \sin\psi + v_A \cos\psi + s\ (\Omega - r_A) \right. \\ & \left. - z(p_A \cos\psi - q_A \sin\psi) \right] + k \left[w_A + \dot{z} + s(p_A \sin\psi \right. \\ & \left. + q_A \cos\psi) \right] \end{aligned} \tag{7S}$$

$$\begin{pmatrix} u_s \\ u_c \\ u_n \end{pmatrix} = T\ \bar{V}_A \tag{8S}$$

$$T = \left[\begin{array}{c|c|c} 1 & 0 & -z' \\ \hline -\theta z & 1 - \dfrac{\theta^2}{2} & -\theta \\ z' & \theta & 1 - \dfrac{\theta^2}{2} \end{array}\right] \qquad (9S)$$

$$\theta = \theta_o - A_{1s} \cos\psi - B_{1s} \sin\psi - \theta_1 \left(\frac{s}{R}\right) + \delta_3\beta \qquad (10S)$$

$$z' = z_o'(s) + z_1'(s)\ \beta \qquad (11S)$$

$$f_n = -\frac{\rho c}{2}\left[(a + \delta_o)\ u_n u_c + \delta_1\ u_n^2\right] \qquad (12S)$$

$$f_c = -\frac{\rho c}{2}\left[(\delta_2 - a)\ u_n^2 + \delta_o\ u_c^2 + \delta_1\ u_n u_c\right] \qquad (13S)$$

$$\begin{pmatrix} p_{xa} \\ p_{ya} \\ p_{za} \end{pmatrix} = T^{-T} \begin{pmatrix} 0 \\ f_c \\ f_n \end{pmatrix} \qquad (14S)$$

$$p_x = p_{xi} + p_{xa} \qquad (15S)$$

$$p_y = p_{ya} \qquad (16S)$$

$$p_z = p_{zi} + p_{za} \qquad (17S)$$

$$F_g = \int_o^R z_1\ p_{za}\ ds - I_1\ g_z - I_2\ (\dot{p} - 2\Omega q)\sin\psi + (\dot{q} + 2\Omega p)\cos\psi\Big] \qquad (18S)$$

$$X_r = \int_0^R p_{xa}\, ds \tag{19S}$$

$$Y_r = \int_0^R p_{ya}\, ds \tag{20S}$$

$$Z_r = \int_0^R p_{za}\, ds - I_1 \left(\frac{F_g}{M_g} - \omega^2 \beta \right) \tag{21S}$$

$$L_r = - \int_0^R z\; p_{ya}\, ds \tag{22S}$$

$$M_r = \int_0^R \left[z\; p_{xa} + s\; p_{za} \right] ds - I_2 \left[\frac{F_g}{M_g} + (\Omega^2 - \omega^2)\beta \right]$$

$$- J_B \left[-2\Omega q \sin\psi + 2\Omega p \cos\psi \right] \tag{23S}$$

$$N_r = - \int_0^R s\; p_{ya}\, ds \tag{24S}$$

$$I_1 = \int_0^R m\; z_1\, ds \tag{25S}$$

$$I_2 = \int_0^R m\; z_1 s\, ds \tag{26S}$$

$$J_B = \int_0^R ms\, ds \tag{27S}$$

3. Solution of the Equations

The spatial integrals (integration from 0 to R) expressed in Eqs. (18S)-(24S) are evaluated using the trapezoidal method. This process is straightforward and requires no expansion here, except to note that for any value of ψ_k, the integrals can be performed when $\beta(\psi_k)$ and $\dot{\beta}(\psi_k)$ are known.

The time integration of Eq. (28) is performed as follows. Given the values of ψ_k, $\beta(\psi_k)$, $\dot{\beta}(\psi_k)$, it is necessary to determine $\beta(\psi_{k+1})$ and $\dot{\beta}(\psi_{k+1})$, where ψ_{k+1} is given by:

$$\psi_{k+1} = \psi_k + \Delta\psi \tag{134}$$

$$\Delta\psi \triangleq 2\pi/(\text{Specified number of azimuthal integration elements}) \tag{135}$$

Since β, $\dot{\beta}$ and ψ are known, the generalized force, F_g, and the blade hub loads can be computed (if the hub loads are required) in a straightforward manner. Assuming F_g constant over azimuthal interval $\Delta\psi$, Eq. (28) becomes

$$\ddot{\beta} + \omega^2\beta = \text{constant} = F_g/M_g \tag{136}$$

This is a total differential equation with constant coefficients, and has the solution

$$\beta(t) = \left[\beta(\psi_k) - \frac{F_g}{M_g\omega^2}\right]\cos\omega t + \left[\frac{\dot{\beta}(\psi_k)}{\omega}\right]\sin\omega t$$

$$+ \frac{F_g}{M_g\omega2} \tag{137}$$

$$\dot{\beta}(t) = -\left[\beta(\psi_k) - \frac{F_g}{M_g\omega^2}\right]\omega\sin\omega t + \left[\dot{\beta}(\psi_k)\right]\cos\omega t \tag{138}$$

where the initial values of β and $\dot{\beta}$ ($\beta(\psi_k)$ and $\dot{\beta}(\psi_k)$) have been used to determine the arbitrary constants in the homogeneous solution of Eq. (136). To determine $\beta(\Psi_{k+1})$ and $\dot{\beta}(\Psi_{k+1})$, set

$$t = \frac{\Delta\psi}{\Omega} \tag{139}$$

Substituting into Eqs. (96) and (136),

$$\beta(\psi_{k+1}) = \left[\beta(\psi_k) - \frac{F_g}{M_g\,\omega^2}\right] K_1 + \left[\frac{\dot{\beta}(\psi_k)}{\omega}\right] K_2 + \frac{F_g}{M_g\,\omega^2} \tag{140}$$

$$\dot{\beta}(\psi_{k+1}) = - \left[\beta(\psi_k) - \frac{F_g}{M_g\,\omega^2}\right] \omega\, K_2 + \left[\dot{\beta}(\psi_k)\right] K_1, \tag{141}$$

where

$$K_1 = \cos\left(\frac{\omega\Delta\psi}{\Omega}\right) = \text{constant} \tag{142}$$

$$K_2 = \sin\left(\frac{\omega\Delta\psi}{\Omega}\right) = \text{constant} \tag{143}$$

The numerical integration technique outlined above provides much more accurate results than the slightly simpler approach (called the Euler method) which uses (136) to solve for $\ddot{\beta}(\psi_k)$, and then calculates $\dot{\beta}(\psi_{k+1})$ and $\beta(\psi_{k+1})$ based on the assumption that $\ddot{\beta}$ is constant over the interval $\Delta\psi$. In fact, the experience with this technique has been very positive, both with respect to the simplicity of the mechanization (resulting in very rapid computation) and with the accuracy.

Knowing $\sin(\psi_k)$ and $\cos(\psi_k)$, the quantities $\sin(\psi_{k+1})$ and $\cos(\psi_{k+1})$ are easily computed by direct substitution of Eq. 134. Making the usual trigonometric expansion,

$$\sin(\psi_{k+1}) = K_3 \sin(\psi_k) + K_4 \cos(\psi_k) \tag{144}$$

$$\cos(\psi_{k+1}) = K_3 \cos(\psi_k) - K_4 \sin(\psi_k) \tag{145}$$

where

$$K_3 = \cos \Delta\psi = \text{constant} \tag{146}$$

$$K_4 = \sin \Delta\psi = \text{constant} \tag{147}$$

The algorithm specified in Eqs. (144) - (147) can be executed much more rapidly than standard sine and cosine subprograms operating with Eq. (134).

Eqs. (198)-(248) are expressions for the rotating load components applied to the rotor shaft by one blade. These components refer to rotor axes. Part I describes how these rotating components can be resolved to a nonrotating axis system and time averaged. The time average, multiplied by the number of rotor blades, represents the rotor loads on the overall vehicle. The resolution and time averaging process is expressed as Eq. (44) in Part I. This equation expresses the averaging in integral form. If the rotating loads are computed at discrete points around the rotor azimuth (as is the case in SWEEP), Eq. (44) takes the form of a summation. This summation, in the notation of Part I, is

$$f_i = \frac{b}{N} \sum_{i=1}^{N} R(\psi_i) f_{ri} \tag{148}$$

where N is the number of azimuth stations used for the time integration in SWEEP f_{ri} is the vector made up of components X_r, Y_r, Z_r, L_r, M_r, N_r for azimuth angle ψ_i, and $R(\psi)$ is the resolution function required to resolve rotor axis system components to hub axis system components. The vector f_i represents rotor shaft loads referred to hub axes, for b rotor blades. Eq. (148) is easily expressed in component form:

$$X_H = \frac{b}{N} \sum_{i=1}^{N} \left(X_{ri} \cos\psi_i + Y_{ri} \sin\psi_i \right) \qquad (149)$$

$$Y_H = \frac{b}{N} \sum_{i=1}^{N} \left(-X_{ri} \sin\psi_i + Y_{ri} \cos\psi_i \right) \qquad (150)$$

$$Z_H = \frac{b}{N} \sum_{i=1}^{N} Z_{ri} \qquad (151)$$

$$L_H = \frac{b}{N} \sum_{i=1}^{N} \left(L_{ri} \cos\psi_i + M_{ri} \sin\psi_i \right) \qquad (152)$$

$$M_H = \frac{b}{N} \sum_{i=1}^{N} \left(-L_{ri} \sin\psi_i + M_{ri} \cos\psi_i \right) \qquad (153)$$

$$N_H = \frac{b}{N} \sum_{i=1}^{N} N_{ri} \qquad (154)$$

Eqs.(149)-(154) are calculated during the regular azimuthal summation (integration) in SWEEP, provided that SWEEP is instructed to compute shaft loads.

4. **The Tip Loss Factor**

Chapter IV of the main text of this report presents the derivation of a tip loss model which has the basic form

$$B = 1 - k\sqrt{T_b'} \tag{155}$$

where k is a constant, and T_b' is the distributed aerodynamic load on a blade at the tip, in lb/ft. This tip loss factor is used to reduce the rotor radius to an "effective radius", R_e :

$$R_e = RB \tag{156}$$

As discussed in Chapter IV, the distributed loading function T_b' used in (155) is calculated assuming zero rotor-induced interference velocity.

Based on the expression of Eq. (155), the following tip loss model is presently incorporated in the MOSTAB-B program

$$B = B_0 - k\sqrt{|p_{za}|} \tag{157}$$

where B_0 is a constant (input) tip loss factor, and k is an input constant. p_{za} is the distributed aerodynamic loading at the blade tip, parallel to the rotor shaft, computed assuming zero rotor induced velocity.

Prior to assembling the aerodynamic integrals in Eqs. (188) - (258) the angle of attack of the blade tip using inertial velocities is computed. $\bar{p}_{za}$ is then calculated based on this "inertial speed" angle of attack. B is computed from (157), and the aerodynamic integrations then proceed with R_e (Eq. (156)) as the upper integral bound in lieu of R.

5. Comments on SWEEP Equations

Eqs. (1S)-(27S) represent the expressions presently included in the SWEEP subroutine. The computational methods specified by Eqs. (140)-(147) and (149)-(154) are also presently incorporated. This very basic set of equations can be easily expanded to include more aerodynamic and dynamic phenomena, if the MOSTAB user requires such additional sophistication for his particular problem. The following list outlines some of the most basic steps that might be taken to expand the present SWEEP routine.

(a) Reverse flow can be accounted for in a very elementary fashion by changing the sign of a (i.e., $C_{l\alpha}$) when the sign of u_c changes. This approach is taken in the earlier NACA rotor analyses (references 7 and 8).

(b) Stall and compressibility drag rise can be accounted for by using Eqs. (12S) and (13S) with suitable variable coefficient models of the form

$$a = a\,(u_n, u_c)$$

$$\delta_o = \delta_o\,(u_n, u_c)$$

$$\delta_2 = \delta_2\,(u_n, u_c)$$

Items (a) and (b) represent relatively simple expansions of the SWEEP equations. More involved expansions can be made, resulting in significant (though not necessarily prohibitive) increases in computer time requirements. The list below represents the expansions that could be made with some effort. Many of the suggestions require expansion of the equations derived in Part III.

(1) Elastic torsional blade deformation - requires expansion of Appendix III. Blade pitching moment inertial and aerodynamic loading functions must be developed.

(2) Additional blade dynamic degrees of freedom requires expansion of Part III. Inplane and torsional (and, of course, additional flapping) modes can be added.

(3) Terms can be added to the dynamic expressions to more suitably account for "load-coupled" rotors (e.g., teetering rotors). See Part III.

(4) Unsteady aerodynamic effects can be added (e.g., hysteral stall, feathering damping moments) - requires expansion of the appendix.

(5) Sophisticated "table-look-up" airfoil data can be incorporated. (This is related to (4), above.)

(6) Restriction of constant rotor speed can be removed, allowing stability derivatives on Ω to be calculated.

Although the expansion areas outlined above can be incorporated in SWEEP, they add additional complexity to the equations. Items (a) and (b) can be included with little additional difficulty. Items (1)-(6) (and probably many other effects) can be added, with additional burden on computational time, input data requirements and basic preliminary analysis (Part III). The equations that now exist in SWEEP provide a very good basis for the flight dynamics analysis, however, and probably should be kept relatively intact (possibly with simple expansion like (a)-(d)) for simplified studies. Other versions of SWEEP could be assembled (e.g. SWEEP1, SWEEP2) involving various degrees of additional complexity. It would be possible to choose any routine from the "SWEEP" library to use with the rest of the basic MOSTAB program. One would choose the version with minimum complexity but with the effects needed for the particular study.

E. Subroutine ROTOR

The subroutine receives the following information when it is called:

(a) The three translational and three rotational airspeed components at the rotor reference point. These velocity components are given in vehicle reference axes (see Part I and Chapter V).

(b) The three translational and three rotational inertial velocity components of the rotor reference point. These components are referred to overall vehicle axes.

(c) The time derivatives of the inertial velocity components described in (b), above.

(d) Three Euler angles, ψ_r, θ_r, φ_r, which are used to rotate vectors expressed in overall vehicle coordinates to a reference system (later to be defined as the hub axis system) conveniently related to the rotor. This local coordinate system is fixed to the nonrotating airframe (it does not move with the rotor or swashplate). The reverse resolution process is also done with these angles.

(e) The control variables associated with the rotor (cyclic pitch and collective pitch, in general).

(f) The physical characteristics of the rotor.

When ROTOR is called the first time to analyze a particular aerodynamic rotor, certain constant terms are computed. Among these constants are I_1, I_2, M_1 and J_B specified by Eqs. (25S) (28S), and K_1, K_2, K_3, K_4 specified by Eqs. (145), (146), (147), and (148). All of these equations appear in the SWEEP section. Also, the initial conditions on the blade state variables ($\beta(0)$ and $\dot{\beta}(0)$) are set to estimated values read into the MOSTAB-B program as data. This initial operation occurs only once per flight condition for every aerodynamic rotor on a vehicle.

As soon as the constant terms are generated, ROTOR calls EULER to rotate all of the vectors (e.g., airspeed, inertial velocity) required for the rotor equations (contained in SWEEP) from overall vehicle coordinates to "hub" coordinates (see Part III for the definition of hub coordinates). The constant angles ψ_r, θ_r, φ_r are used for this rotation. If the particular

rotor being analyzed has rigid blades, ROTOR calls SWEEP once to determine the necessary hub loads. EULER is then called (by ROTOR) to rotate these loads from hub axes to overall vehicle axes. For the rotor with rigid blades, the process of load computation is complete at this point and ROTOR returns control to FORCE.

If the rotor being studied has flexible blades, ROTOR determines the blade initial conditions for the particular trim solution being sought by MOSTAB-B. The process used to determine this initial condition is discussed in detail in Part I. In the notation of Part I, SWEEP is called with the initial blade state variables, $\zeta_e(0)$, set to the estimated values read into MOSTAB-B. SWEEP returns to value $\zeta_o(2\pi)$ (i.e., the angles β and $\dot{\beta}$ at $\psi = 2\pi$). The matrix Z_{ZO} is generated by perturbing the elements in the estimated initial condition column (one at a time) and generating the resulting perturbations in blade final conditions using SWEEP. During these blade motion calculations, the shaft loading option in SWEEP is suppressed to save computer time.* The corrected blade initial conditions are found by solving Eq. (54) of Part I, and SWEEP is called again using this corrected initial condition. The option to compute shaft loads (in hub coordinates) is exercised in SWEEP during this last call. These loads are then rotated to overall vehicle coordinates by EULER. The load computing process thus completed, ROTOR returns control to FORCE.

An option is provided in ROTOR to suppress computation of the blade motion gradient matrix Z_{ZO}. This option is used to save influence on the corrected blade initial conditions. When this option is exercised, events proceed as discussed above for the flexible bladed rotor, except that the last value of Z_{ZO} computed for the particular aerodynamic rotor under study is used. Instructions to generate a new Z_{ZO} are suppressed. The conditions that cause re-computation of Z_{ZO} to be unnecessary are discussed in Part I.

---

* The loads applied to the shaft need to be computed only after the blade initial conditions are known. When SWEEP is being used to determine blade motion, the shaft load calculation is incorrect, since the proper initial conditions are not available.

II.3 VEHICLE GEOMETRY

Preceding sections of this work have been involved with the equations which ultimately compute the vehicle force column, f, as represented functionally by Eq. (2) in Part I. This section continues the development of specific expressions for the functional equations in Part I.

Eqs. (3) and (4) in Part I suffice to define the geometric matrices L and G. L sums the loads of all of the vehicle elements to a final six element load column p. G converts overall vehicle motions (as represented by three translational and three rotational inertial velocity components), to the inertial velocity column v_I. The elements of v_I represent the inertial velocity components of all the vehicle elements. It will be shown subsequently that L is simply the transpose of G. This fact saves core space when MOSTAB is used, since only one geometric matrix (either G or L) can be used for both - with suitable adjustments in computer logic of course.

Consider vehicle element i. The reference point of element i can be located with respect to the overall vehicle reference point by a vector d_i. Expressing d_i in component form,

$$d_i = \hat{i}\, x_i + \hat{j}\, y_i + \hat{k} z_i \tag{158}$$

The unit vectors $\hat{i}$, $\hat{j}$, $\hat{k}$ refer to overall vehicle coordinates. In Part I, the symbol s is used to represent the six-element inertial velocity column for the flight vehicle as a whole. s can be split into two vectors:

$$V = \hat{i}u + \hat{j}v + \hat{k}w \tag{159}$$

$$\omega = \hat{i}p + \hat{j}q + \hat{k}r \tag{160}$$

Since d_i is the vector which locates the aircraft element reference point, i, with respect to vehicle axes, the translational velocity of i in vector form is

$$V_i = V + \omega \times d_i \tag{161}$$

where the (x) symbol denotes the vector cross product. Since $\dot{d}_i = 0$ (i.e., d_i is a constant vector in vehicle coordinates),

$$\omega_i = \omega \tag{162}$$

Eqs. (161) and (162) can be written in component form:

$$u_i = u + qz_i - ry_i \tag{163}$$

$$v_i = v + rx_i - pz_i \tag{164}$$

$$w_i = w + py_i - qx_i \tag{165}$$

$$p_i = p \tag{166}$$

$$q_i = q \tag{167}$$

$$r_i = r \tag{168}$$

Clearly, Eqs. (163) - (168) are the component forms of Eq. (4) of Part I, for a single vehicle element. The G matrix for element i, is thus given by the expression

$$G_i(x_i, y_i, z_i) = \begin{bmatrix} 1 & & & & z_i & -y_i \\ & 1 & & -z_i & & x_i \\ & & 1 & y_i & -x_i & \\ & & & 1 & & \\ & & & & 1 & \\ & & & & & 1 \end{bmatrix} \tag{169}$$

unlabeled elements are zero

The overall vehicle geometric matrix, G, is assembled by stacking all of the submatrices G_i into one matrix having dimension 6NX6, where N is the number of vehicle elements. The submatrices, G_i, are stacked one on top of the other in G, starting at the top. (The order of submatrices G in G is defined by the definition of the column, v_I, in Part I).

$$G = \begin{bmatrix} G_1 \\ \hdashline G_2 \\ \hdashline \vdots \\ \hdashline G_N \end{bmatrix} \tag{170}$$

The array, G, as defined by Eqs. (169) and (170), is the matrix presently mechanized in MOSTAB. If L is required, G^T is used. To show that $L = G^T$, the general expression for L is now developed. Let the force and moment vectors developed by element i be denoted F_i and M_i respectively. These vectors are applied to the aircraft at element i's reference point. They contribute an effective force and moment at the vehicle reference point.

$$F_{vi} = F_i \tag{171}$$

$$M_{vi} = M_i - d_i \times F_i \tag{172}$$

The sign is negative in (172) because $-d_i$ locates the vehicle reference point with respect to the element reference point. Writing (171) and (172) in component form,

$$X_{vi} = X_i \tag{173}$$

$$Y_{vi} = Y_i \tag{174}$$

$$Z_{vi} = Z_i \tag{175}$$

$$L_{vi} = L_i + y_i Z_i - z_i Y_i \tag{176}$$

$$M_{vi} = M_i + z_i X_i - x_i Z_i \tag{177}$$

$$N_{vi} = N_i + x_i Y_i - y_i X_i \tag{178}$$

Eqs. (173) - (178) simply represent the component form of Eq. (3) of Part I for a single vehicle element. Thus, the L matrix for a single element is given by

$$L_i(x_i, y_i, z_i) = \begin{bmatrix} 1 & & & & & \\ & 1 & & & & \\ & & 1 & & & \\ & -z_i & y_i & 1 & & \\ z_i & & -x_i & & 1 & \\ -y_i & x_i & & & & 1 \end{bmatrix} \tag{179}$$

unlabeled elements are zero

The overall vehicle matrix, L, is assembled by placing all of the submatrices L_i into one matrix having dimensions 6X6N, where N is the number of vehicle elements. The submatrices, L_i, are placed side by side in L, starting from the left. (The order of submatrices L_i in L is defined this way because of the definition of column f in Reference 1).

$$L = \left| \; L_1 \quad L_2 \quad \cdots\cdot \quad L_N \; \right| \tag{180}$$

Clearly,

$$L_T = \begin{bmatrix} L_1^T \\ \hdashline L_2^T \\ \hdashline \\ \hline L_N^T \end{bmatrix} \tag{181}$$

from inspection of (180). Also, for an element i,

$$L_i^T = G_i \tag{182}$$

from inspection of Eqs. (169) and (179). It follows directly that

$$G = L^T, \tag{183}$$

which is the desired result.

II.4 INTERFERENCE VELOCITY COMPUTATION (Subroutine WASH)

Continuing to Eq. (7) of Part I, the interference velocity column, w, is expressed in a functional form:

$$w = w\ (f,\ v_A, v_I, \dot{v}_I, c,\ K_\ell, \ell = 1,2 \cdots) \tag{7}$$

The purpose of subroutine WASH is to produce the column w, given the quantities shown as arguments in Eq.(7). Many models for interference velocity have been proposed and used. These models are functions of the vehicle type, flight regime, etc. Although a rather general (and classical) interference velocity model is presently used in MOSTAB, subroutine WASH will undoubtedly go through many phases of refinement as MOSTAB is used to study various kinds of flight vehicles.

The interference velocity model presently incorporated in MOSTAB will now be discussed. Define the six element column d_i whose elements are made up of the three translational and three rotational interference velocities at element reference point i. The velocity d_i is caused only by element i (i.e., it contains no interference velocity effects from elements near element i). Eq.(184)gives a general expression for d_i:

$$d_i = \frac{-1}{2\rho\,|V_{ATi}|}\ A_i\ f_i \tag{184}$$

The symbol $|V_{ATi}|$ represents the scalar magnitude of the translational airspeed at element i. Three elements of v_A represent the components of translational airspeed at the reference point, i. The square root of the sum of the squares of these components is $|V_{ATi}|$. The symbol A_i represents a 6X6 square array which is input to MOSTAB. Its elements have units of 1/area, and represent the inverses of the characteristic areas of element i. The six-element load column produced by element i is F_i. Both d_i and f_i are referred to overall vehicle coordinates.

Eq. (2) is essentially a generalized form of the Glauert expression for lifting rotors. This classical expression for downwash of a lifting rotor takes the scalar form:

$$\nu = \frac{T}{2\pi\ R^2\ \rho V'}$$

where ν is rotor downwash, T is thrust, R is the rotor radius and V' is the magnitude of translational airspeed at the rotor (i.e., $V' = |V_{ATi}|$ for the rotor). Clearly,(185)is a specific form of (184)when one element of A_i is $1/\pi R^2$.

In substantiating the use of (185), the arguments show that this equation holds for rotors in axial flight (as derived by either momentum or vortex theory) and for elliptical wings

in forward flight. Eq.(185)has enjoyed rather broad usage in the analysis of wings and rotors in forward flight.

Now form the total 6N element column d by inserting submatrices d_i into d, one on top of the other, starting at the top. At the present time, a constant coupling matrix, X, is input to MOSTAB, to represent the interelement induced velocity interference:

$$w = Xd \tag{186}$$

X is a 6NX6N matrix. Eventually, X should be made a function of d and v_A, to account for wake angles, etc.

II.5 CONTROL SYSTEM (Subroutine CONTRL)

The operation of this subroutine is characterized by Eq. (11) of Part I.

$$c_t = c(t, \text{known constraints, known constants})$$

At the present time, CONTRL is simply a logic routine which determines which elements of c_t relate to elements of t, and which are constrained by the trim problem definition. One-to-one relationships are used between c_t and t.

If a control system is used with MOSTAB, the equations representing the system would be included in CONTRL. For example, if t had such elements as cyclic stick position, collective stick position, etc., CONTRL would determine the aircraft element-oriented control settings in c_t (e.g., cyclic pitch angle, collective pitch angle) by using suitable equations for the linkages between the control sticks and the rotor(s).

II.6 INERTIAL VELOCITY (Subroutine VELCTY)

Eq. (12) of Part I indicates the dependence of the trim inertial velocity column, s_t, on the trim variable column, t, and the constraints of the trim problem.

$$s_t = s_t \text{ (t, known constraints)} \quad (12)$$

The elements of t, and the constraints input to MOSTAB which define the trim problem, are listed below. Some of the items in this list are not used by VELCTY, but are required by the subroutine FCERQD. The appropriate items in the list required by FCERQD will be considered in the section dealing with that subroutine.

(a) Vehicle weight, W

(b) Overall vehicle axis system coordinates of the aircraft's center of gravity: x_{cg}, y_{cg}, z_{cg}

(c) Speed of the vehicle in space, denoted V in this analysis

(d) Air density, ρ

(e) Turning rate ($\dot{\Psi}$ in classical airplane notation for the yawing Euler angle)

(f) Pitch rate (either $\dot{\theta}$ or q can be specified, by option index, to represent pitch rate)

(g) Roll rate (either $\dot{\phi}$ or p can be specified, by option index, to represent roll rate)

(h) Rate of climb ($\dot{h}$)

(i) The inertia tensor of the vehicle referred to overall vehicle axes [I]

(j) The sideslip velocity (v)

(k) The pitch Euler angle for vehicle axes (θ)

(l) The roll Euler angle for vehicle axes (ϕ)

The reader is referred to Reference 9 for the classical airplane dynamics analysis. The notation used in Reference 9 will be used here.

The problem encountered by VELCTY can be stated in a mathematical format as follows:

Given: $\Theta, \Phi, v, (p \text{ or } \dot{\Phi}), (q \text{ or } \dot{\Theta}), \dot{\Psi}, \dot{h}$ and V

Determine: the inertial velocity components u, v, w, p, q, r - referred to overall vehicle axes

The component v is given and requires no more consideration. Figure 9 shows some of the basic notation required to calculate u and w from the given information. The total inertial velocity vector of the aircraft reference axes in space is shown as $\bar{V}$.

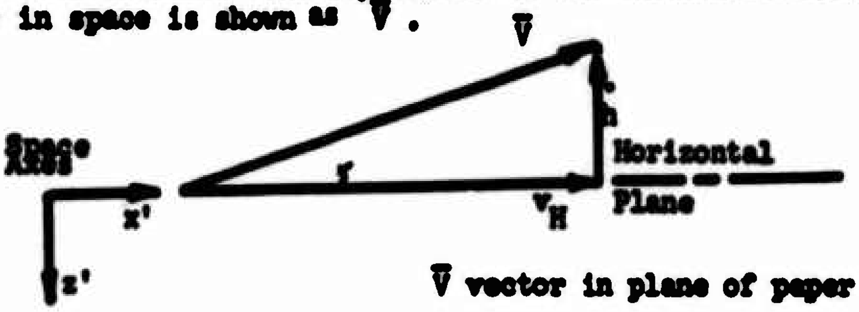

Figure 9. Velocity Resolution.

The horizontal plane can be defined as a plane normal to the action of gravity. Since the magnitude of $\bar{V}$ (which is V) and $\dot{h}$ are known quantities, the ground speed v_H can be computed directly

$$v_H = \sqrt{V^2 - \dot{h}^2} \geq 0 \tag{187}$$

Now assign a "space" axis system as shown in Figure 9 that is, a coordinate system with its z axis in the direction of gravity, and its xz plane containing the vector $\bar{V}$. Three Euler angles, Ψ, Θ and Φ can be used to rotate these space axes to aircraft axes. The only unknown of these Euler rotations is Ψ, since Θ and Φ are given. Reference 9 shows the equations required to express the aircraft's velocity, $\bar{V}$, in spatial coordinates, in terms of the velocity expressed in aircraft axes and the Euler angles Ψ, Θ and Φ. Expressed in matrix form, these equations are

$$\begin{pmatrix} dx'/dt \\ dy'/dt \\ dz'/dt \end{pmatrix} = \left[\begin{array}{c:c:c} \cos\Theta\cos\Psi & \sin\Phi\sin\Theta\cos\Psi - \cos\Phi\sin\Psi & \cos\Phi\sin\Theta\cos\Psi + \sin\Phi\sin\Psi \\ \cos\Theta\sin\Psi & \sin\Phi\sin\Theta\sin\Psi + \cos\Phi\cos\Psi & \cos\Phi\sin\Theta\sin\Psi - \sin\Phi\cos\Psi \\ -\sin\Theta & \sin\Phi\cos\Theta & \cos\Phi\cos\Theta \end{array}\right] \begin{pmatrix} u \\ v \\ w \end{pmatrix} \tag{188}$$

Eq.(188) holds for any space axis system if z' lies parallel to the action of gravity. For the system of Figure 1, the spatial velocity components of the aircraft velocity vector $\bar{V}$ are

$$dx'/dt = v_H = \sqrt{V^2 - \dot{h}^2} \tag{189}$$

$$dy'/dt = 0 \tag{190}$$

$$dz'/dt = -\dot{h} \tag{191}$$

The Eulerian rotational matrix of Eq. (188) can be inverted by transposition to solve for u, v and w in terms of dx'/dt, dy'/dt and dz'/dt. The solution for v achieved from this process

$$v = \sqrt{V^2-\dot{h}^2}\left[(\sin\Phi\sin\Theta)\cos\Psi-(\cos\Phi)\sin\Psi\right]-(\sin\Phi\cos\Theta)\dot{h} \tag{192}$$

Since v is given, the only unknown in Eq.(192) is Ψ.

Before proceeding with the solution of (192), define the known quantity, C:

$$C \triangleq \frac{v + \sin\Phi\cos\Theta\dot{h}}{\sqrt{V^2 - \dot{h}^2}} \tag{193}$$

With this definition, Eq.(192) is written simply as

$$(\sin\Phi\sin\Theta)\cos\Psi-(\cos\Phi)\sin\Psi = C \tag{194}$$

Figure 10 is a graphic display of Eq. (194):

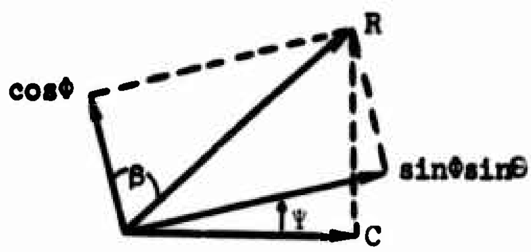

Figure 10. Graphic Display of Eq. 194

Two vectors (at right angles to each other) are given actual lengths of $\cos\Phi$ and $\sin\Phi\sin\Theta$ as shown in Figure 10. Clearly, the vector C is related to Ψ, Θ and Φ in accordance with Eq. (10). The vector R is the resultant of $\cos\Phi$ and $\sin\Phi\sin\Theta$, and is defined positive at all times. The expression for R can be written immediately:

$$R = \sqrt{\cos\Phi^2 + (\sin\Phi\sin\Theta)^2} \tag{195}$$

From inspection of Figure 10, one sees that a value of Ψ exists that satisfies the diagram only if

$$R \geq |C| \tag{196}$$

If condition 196 is not met, the given values of Φ, Θ, v, V and $\dot{h}$ (which make up R and C) are impossible. The action taken by VELCTY in this event will be discussed later. For the time being, assume that condition 196 is fulfilled. In this case, two values of Ψ will work to satisfy Figure 10:

$$R \cos\left[\pm(\Psi + \pi/2 - \beta)\right] = C \qquad \text{or}$$

$$\Psi = \beta - \pi/2 \pm \cos^{-1}(C/R) \tag{197}$$

Since $\cos\Psi$ and $\sin\Psi$ are required, (11) can be used to get

$$\cos\Psi = \cos\left[\beta - \pi/2 \pm\cos^{-1}(C/R)\right] = \sin\left[\beta \pm\cos^{-1}(C/R)\right] \tag{198}$$

$$\sin\Psi = \sin\left[\beta - \pi/2 \pm\cos^{-1}(C/R)\right] = -\cos\left[\beta \pm\cos^{-1}(C/R)\right] \tag{199}$$

Using the trigonometric relationship

$$\sin\left[\cos^{-1}X\right] = \pm\sqrt{1 - X^2} \tag{200}$$

expressions (198) and (199) are reduced to the form

$$\cos\Psi = \pm\sqrt{1-(C/R)^2}\cos\beta + (C/R)\sin\beta \tag{201}$$

$$\sin\Psi = -(C/R)\cos\beta \pm\sqrt{1-(C/R)^2}\sin\beta \tag{202}$$

Eqs. (201) and (202) can be simplified by noting that

$$\cos \beta = (\cos \Phi)/R \tag{203}$$

and

$$\sin \beta = (\sin \Theta \sin \Phi)/R \tag{204}$$

from inspection of Figure 10. Substituting (203) and (204) into (201) and (202),

$$\cos \Psi = \left[\pm \sqrt{1 - \left(\frac{C}{R}\right)^2} \cos \Phi + \left(\frac{C}{R}\right) \sin \Phi \sin \Theta \right] \left(\frac{1}{R}\right) \tag{205}$$

$$\sin \Psi = \left[- \left(\frac{C}{R}\right) \cos \Phi \pm \sqrt{1 - \left(\frac{C}{R}\right)^2} \sin \Phi \sin \Theta \right] \left(\frac{1}{R}\right) \tag{206}$$

The requirement that $R^2 \geq C^2$ is easily seen in Eqs. (205) and (206).

The ($\pm$) sign on the radicals in Eqs. (205) and (206) requires special attention at this point. To clarify the meaning of these sign options, solve for u, v, w using Eq.(188) in terms of dx'/dt and dz'/dt. All that is required is the transpose of the square matrix in(188)(which is, of course its inverse since the matrix is an Eulerian rotational matrix). Noting that dy'/dt = 0 from (190), the second column of this transpose may be omitted. Using Eqs.(189) and (191) to substitute for dx'/dt and dz'/dt, the solution for u, v and w takes the matrix form:

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \left[\begin{array}{l|l} \cos \Theta \cos \Psi & -\sin \Theta \\ \sin \Phi \sin \Theta \cos \Psi - \cos \Phi \sin \Psi & \sin \Phi \cos \Theta \\ \cos \Phi \sin \Theta \cos \Psi + \sin \Phi \cos \Psi & \cos \Phi \cos \Theta \end{array}\right] \begin{pmatrix} \sqrt{V^2 - \dot{h}^2} \\ -\dot{h} \end{pmatrix} \tag{207}$$

The second equation in (207) yields the identity $v = v$ if Eqs. (205) and (206) are substituted for $\cos \Psi$ and $\sin \Psi$, verifying the correctness of Eqs. (205) and (206). To grasp the meaning of the $\pm$ sign on the radical, observe the first equation in (207) for u, and imagine a vehicle in level, coordinated flight ($\dot{h} = \Phi = \Theta = v = 0$). Under these conditions, C is zero (as can be seen from Eq. (193) and R is unity (from Eq. (195). Solving for u (without altering the radical in Eq. (205),

$$u = V \cos \Psi = V \left(\pm \sqrt{1}\right) \tag{208}$$

If the + sign is chosen on the radical, flight is forward, corresponding to a + u. The (-) sign defines the flight as backward, a condition which is not only possible, but important for helicopters. Thus, an option is included in VELCTY (by index) to assign either the (+) or (-) sign on the radicals in Eqs. (205) and (206).

(+) ⟶ forward flight
(-) ⟶ backward flight

If condition (196) is fulfilled, Eqs. (205) and (206) can be used to determine $\cos \Psi$ and $\sin \Psi$, for substitution into (207) to yield u and w. If condition (196) is not met, the given quantities are incompatible. In this case, VELCTY fails to use the given sideslip velocity, v, and calculates a new value from Eq. (207). The value of v is relatively arbitrary at this point. VELCTY sets $|C| = R$ and calculates $\cos \Psi$ and $\sin \Psi$ from Eqs. (205) and (206) on this basis. Of course, C can be either (+) or (-) R, so VELCTY uses the forward/backward index to specify the sign on C.

Forward flight: $C = +R$
Backward flight: $C = -R$

After $\cos \Psi$ and $\sin \Psi$ are calculated, u, v and w can be computed using (207). In this case, v will be different from the given value.

Table III is presented to specify the action of VELCTY for the contingencies on the sizes of R and C (discussed above), and other limiting cases which arise.

TABLE III. OPTIONS FOR SUBROUTINE VELCTY		
Case No.	Condition	VELCTY Action
I	$V = 0$	$u = v = w = 0$
II	$\|\dot{h}\| = V$	$u = - \dot{h} \sin \Theta$, $v = \dot{h} \sin \Phi \cos \Theta$ $w = \dot{h} \cos \Phi \cos \Theta$
III	$\|C\| \leq R$	Calculate $\sin \Psi$ and $\cos \Psi$ from equations (18) and (19), and u, v and w from (20). v will be the same as the given value in this case - providing a check on the VELCTY computation.
IV	$\|C\| > R$	Set $C = + R$ if flight - forward, $C = - R$ if flight backward. Compute $\sin \Psi$ and $\cos \Psi$ from (18) and (19), and solve for u, v and w from (20).

Rotational velocities p, q and r are now determined.

Sometimes p and q are specified before VELCTY is called. If $\dot{\Theta}$ and $\dot{\Phi}$ are specified in lieu of either p or q, the following formulas are used (taken directly from Reference 9):

$$p = \dot{\Phi} - \dot{\Psi} \sin\Theta \tag{209}$$

$$q = \dot{\Theta} \cos\Phi + \dot{\Psi} \cos\Theta \sin\Phi \tag{210}$$

The yawing rate, r, is obtained directly from given information by slightly rearranging an equation of Reference 9.

$$r = (\dot{\Psi} \cos\Phi - q \sin\Phi) \sec\Phi \tag{211}$$

II.7 REQUIRED TRIM LOADS (Subroutine FCERQD)

Eq. (14) of Part I expresses the functional relationship between the required trim force-moment column, r, and the trim variables and constraints:

$$r = r(t, \text{ problem constraints, constants}) \tag{212}$$

The elements of r are the three force and three moment components of the required trim aerodynamic load. These load components are taken at the overall vehicle reference point, and are referred to vehicle axes. r can be determined from items a-e listed in the VELCTY section of this report. The notation of Reference 9 was used in the VELCTY section, and will also be used here.

Before r can be determined, s must be known. Thus, VELCTY must be called before FCERQD may be called, for a given set of trim requirements.

Consider a center of gravity (cg) coordinate system which is parallel to the vehicle axes but whose origin lies at the vehicle cg. A G matrix can be defined (see the Vehicle Geometry section) which can be used to determine the inertial velocity components of the cg.

$$s_{cg} = G_{cg} s \tag{213}$$

Eq. (2) can be expanded into component form. The resulting component equations are analogous to Eqs. (6) - (11) in the Vehicle Geometry section:

$$u_{cg} = u + q\, z_{cg} - r\, y_{cg} \tag{214}$$

$$v_{cg} = v + r\, x_{cg} - p\, z_{cg} \tag{215}$$

$$w_{cg} = w + p\, y_{cg} - q\, x_{cg} \tag{216}$$

$$p_{cg} = p \tag{217}$$

$$q_{cg} = q \tag{218}$$

$$r_{cg} = r \tag{219}$$

(Note that the r used in symbolic Eq. (1) has no connection with the r yawing rate).

The equations of motion for a rigid-body flight vehicle are derived in Reference 9. These equations refer to a body axis system located at the aircraft's cg. Part I describes the

definition of trim used in MOSTAB: $\dot{s} = 0$. Thus, for trim, $\dot{u}_{cg} = \dot{v}_{cg} = \dot{w}_{cg} = \dot{p}_{cg} = \dot{q}_{cg} = \dot{r}_{cg} = 0$. The rigid body equations, constrained by the trim definition used in MOSTAB, can be written

$$X_{cg} = W \sin\theta + \frac{W}{g}\left[q_{cg}\, w_{cg} - r_{cg}\, v_{cg}\right] \tag{220}$$

$$Y_{cg} = - W \cos\theta \sin\phi + \frac{W}{g}\left[r_{cg}\, u_{cg} - p_{cg}\, w_{cg}\right] \tag{221}$$

$$Z_{cg} = - W \cos\theta \cos\phi + \frac{W}{g}\left[p_{cg}\, v_{cg} - q_{cg}\, u_{cg}\right] \tag{222}$$

$$L_{cg} = q_{cg}\, h_{zcg} - r_{cg}\, h_{ycg} \tag{223}$$

$$M_{cg} = r_{cg}\, h_{xcg} - p_{cg}\, h_{zcg} \tag{224}$$

$$N_{cg} = p_{cg}\, h_{ycg} - q_{cg}\, h_{xcg} \tag{225}$$

$$h_{xcg} = I_{xx}\, p_{cg} - I_{xy}\, q_{cg} - I_{xz}\, r_{cg} \tag{226}$$

$$h_{ycg} = - I_{yx}\, p_{cg} + I_{yy}\, q_{cg} - I_{yz}\, r_{cg} \tag{227}$$

$$h_{zcg} = - I_{zx}\, p_{cg} - I_{zy}\, q_{cg} + I_{zz}\, r_{cg} \tag{228}$$

Eqs. (220)-(228) define the required force and moment expressions for vehicle trimmed conditions. These required trim loads are referred to an axis system fixed to the aircraft, with origin at the center of gravity. MOSTAB needs the loads required for trim to be expressed with respect to vehicle axes (recall that "vehicle axes" are fixed to the aircraft "in a convenient position" generally not the cg). The loads expressed by Eqs. (220)-(225) are easily translated to vehicle reference axes.

Define $\bar{d}$ as a vector locating the cg with resptect to the origin of the vehicle axes. In component form,

$$\bar{d} = \hat{i}\, x_{cg} + \hat{j} y_{cg} + \hat{k}\, z_{cg} \quad \text{expressed in vehicle axis system coordinates} \tag{229}$$

Let $\bar{F}_{cg}$ and $\bar{\Gamma}_{cg}$ be the required force and moment vectors at the cg. Components of $\bar{F}_{cg}$ are given by Eqs.(220) - (222), and components of Γ_{cg} are given by Eqs.(223) - (225). Now let $\bar{F}$ and $\bar{\Gamma}$ be the loads required at the origin of the vehicle axes, which produce the equivalent loading system F_{cg} and Γ_{cg}. Clearly,

$$\bar{F}_{cg} = \bar{F} \tag{230}$$

$$\bar{\Gamma}_{cg} = \bar{\Gamma} - \bar{d} \times \bar{F} \tag{231}$$

Solving (230) and (231) for $\bar{F}$ and $\bar{\Gamma}$,

$$\bar{F} = \bar{F}_{cg} \tag{232}$$

$$\bar{\Gamma} = \bar{\Gamma}_{cg} + \bar{d} \times \bar{F}_{cg} \tag{233}$$

Eqs.(232) and (233) are expanded in component form below:

$$X = X_{cg} \tag{234}$$

$$Y = Y_{cg} \tag{235}$$

$$Z = Z_{cg} \tag{236}$$

$$L = L_{cg} + y_{cg} Z_{cg} - z_{cg} Y_{cg} \tag{237}$$

$$M = M_{cg} + z_{cg} X_{cg} - x_{cg} Z_{cg} \tag{238}$$

$$N = N_{cg} + x_{cg} Y_{cg} - y_{cg} X_{cg} \tag{239}$$

Eqs. (234) - (239) express the components of the trim-force column (Eq. 212). The set of Eqs. (214) - (219) - (220) - (228) and (234) - (239) are those presently mechanized in FCERQD. These expressions suffice to define the functional relationship (212).

II.8 GENERAL MATRIX OPERATION SUBROUTINES

A. Subroutine EULER

Reference 9 documents the standard Eulerian coordinate system transformation. The method is presented here without derivation, since it is a very standard procedure.

Given a vector V expressed in coordinates of some orthogonal axis system a,

$$V = i_a\, v_{xa} + j_a\, v_{ya} + k_a\, v_{za} \tag{240}$$

It is necessary to express V in the coordinates of axis system b.

$$V = i_b\, v_{xb} + j_b\, v_{yb} + k_b\, v_{zb} \tag{241}$$

Three angles, ψ, θ, φ, exist such that the components of V in b coordinates can be calculated as functions of the 'a' coordinates. In matrix form, this relationship is expressed:

$$\begin{pmatrix} v_{xb} \\ v_{yb} \\ v_{zb} \end{pmatrix} = R(\varphi)\; R(\theta)\; R(\psi) \begin{pmatrix} v_{xa} \\ v_{ya} \\ v_{za} \end{pmatrix} \tag{242}$$

The quantities $(R(\psi), R(\theta)$, and $R(\varphi))$ are 3 x 3 arrays called rotational matrices. These arrays are defined as follows:

$$R(\psi) = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{243}$$

$$R(\theta) = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \tag{244}$$

$$R(\varphi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi & \sin\varphi \\ 0 & -\sin\varphi & \cos\varphi \end{bmatrix} \tag{245}$$

The order of rotation is relevant, meaning that the order of multiplication in Eq.(242) is relevant. An important property of an Eulerian rotational matrix is that its inverse is equal to its transpose. Another important property is that its transpose is equal to the untransposed matrix with the sign of the angle changed. To show these important properties in mathematical form, refer back to the arbitrary vector V. Denote the matrix column made up of the elements of V expressed in coordinates a as V_a. V expressed as a column in b coordinates is denoted as V_b. With this notation, (242) can be written

$$V_b = R(\varphi)\ R(\theta)\ R(\psi)\ V_a \tag{246}$$

Premultiplying (247) through consecutively by $R^{-1}(\varphi)$, $R^{-1}(\theta)$, $R^{-1}(\psi)$ and transposing,

$$V_a = R^{-1}(\psi)R^{-1}(\theta)R^{-1}(\varphi)V_b \tag{247}$$

The properties of rotational matrices discussed above allow (247) to be written in two other forms:

$$V_a = R^T(\psi)R^T(\theta)R^T(\varphi)V_b \quad \text{and} \tag{248}$$

$$V_a = R(-\psi)R(-\theta)R(-\varphi)V_b \tag{249}$$

Subroutine EULER performs operations (246) or (249), as specified by an option index. An additional option can be exercised in EULER. Computing sine and cosine functions digitally as required in Eqs. (243), (244), (245) is a relatively time-consuming process. If the Euler angle (e.g., ψ) used to assemble the rotational matrix is small, the following trigonometric approximations are accurate:

$$\sin\psi \cong \psi \tag{250}$$

$$\cos\psi \cong 1 - 1/2\ \psi^2 \tag{251}$$

By option. Euler uses Eqs.(250) and (251) to generate sine and cosine functions. instead of using the computer library trigonometric functions.

B. Standard Matrix Subroutines

The list below shows the general matrix subroutines (other than EULER) presently incorporated in MOSTAB. These subroutines are so general in purpose that no further explanation is required. The input/output formats are described, by comment, in the FORTRAN listings.

Subroutine Name	Function
MATINV	Matrix inversion
MTXADD	Matrix addition or subtraction
MTXMPY	Matrix multiplication

II.9 STABILITY DERIVATIVE MATRIX RESOLUTION

The stability derivative expression is given by Eq. (10) of Part I.

$$\Delta p = P_s \, \Delta s + P_{\dot{s}} \, \Delta \dot{s} + P_c \, \Delta c \qquad (252)$$

The column Δp is the perturbation load column expressed in overall vehicle coordinates. Δs is the perturbation column in vehicle inertial velocity components. $\Delta \dot{s}$ is the time derivative of Δs. Δc is the control perturbation column. Note that Δp represents loads applied at the overall vehicle reference point, and Δs and $\Delta \dot{s}$ represent inertial velocity and acceleration of this reference point.

The position of the cg (with respect to the overall vehicle coordinate system) is defined by three input dimensions: x_{cg}, y_{cg}, z_{cg}. Define a cg coordinate system parallel to vehicle coördinates but with its origin at the cg. One can say that the vehicle asix system origin is located with respect to cg axes by coordinates $-x_{cg}$, $-y_{cg}$, $-z_{cg}$, in cg coordinates. In the Geometry section of this appendix, a matrix, G_1, was defined which can be used to relate motions of the vehicle axes knowing cg motions:

$$s = G_{cg}(-x_{cg}, -y_{cg}, -z_{cg}) s_{cg} \qquad (253)$$

A matrix L_1 was also defined in the Geometry section which can be used to calculate the effective loading system at one point of the vehicle, knowing the loads at another point. The matrix L_1 is used in the following equation to determine cg loads knowing overall vehicle reference point loads.

$$p_{cg} = L_{cg}(-x_{cg}, -y_{cg}, -z_{cg}) p \qquad (254)$$

It was shown in the Geometry portion of this appendix that

$$L_{cg} = G_{cg}^{\;T} \qquad (255)$$

Eqs. (253), (254), and (255) can be used to eliminate Δp, Δs, and $\Delta \dot{s}$ from Eq. (252). The result is

$$\Delta p_{cg} = \left[G_{cg}^{\ T}\ P_s\ G_{cg}\right] \Delta s_{cg} + \left[G_{cg}^{\ T}\ P_{\dot{s}}\ G_{cg}\right] \Delta \dot{s} \tag{256}$$

$$+ \left[G_{cg}^{\ T}\ P_c\right] \Delta c$$

The equation for G is taken directly from the Geometry portion of this Part II with the substitutions $x_i = -x_{cg}$, $y_i = -y_{cg}$, $z_i = -z_{cg}$.

$$G_{cg} = \begin{bmatrix} 1 & & & & -z_{cg} & +y_{cg} \\ & 1 & & +z_{cg} & & -x_{cg} \\ & & 1 & -y_{cg} & x_{cg} & \\ & & & 1 & & \\ & & & & 1 & \\ & & & & & 1 \end{bmatrix} \tag{257}$$

unlabelled elements are zero

Stability axes (sa) are defined as follows:

a) The origin of the stability axis system lies at the aircraft's cg.

b) The cg's inertial velocity components along the y_{sa} and z_{sa} axes are zero.

These two requirements on the position of the stability axis system still do not completely define the position of the axes. Once x_{sa} points in the direction of the aircraft cg's inertial velocity vector, the stability axes can be rotated about x_{sa} to an infinite number of different orientations on the airplane, without violating either of the constraints (a) or (b). The problem seldom arises on fixed-wing airplanes that are usually trimmed with zero sideslip angle. In this case, the stability axes are usually removed from the usual vehicle reference axes by a Eulerian pitch angle (rotation about y_{sa} only).

A third constraint is added to the definition of the stability axes here, to uniquely define their position. This constraint is arbitrary, but seems to be along the lines usually taken in stability analyses.

(c) The z_{sa} lies in the cg axis system's $x_{cg} - z_{cg}$ plane.

The $x_{cg} - z_{cg}$ plane of the cg axis system usually lies parallel to a plane of symmetry of the aircraft. In this case, Constraint (c) above, requires the z_{sa} axis to be parallel to the aircraft's plane of symmetry.

The inertial velocity components of the vehicle, expressed in cg coordinates, must be known before the cg-to-stability axes transformation matrix can be derived. (This fact will be seen later, when this transformation process is developed). Since the cg is defined with respect to overall vehicle axes by coordinates x_{cg}, y_{cg}, z_{cg}, the matrix G_1 (derived in the Geometry section of this report) can be used directly to define the vehicle's inertial velocity components.

$$s_{cg} = G_{cg}\,(x_{cg}, y_{cg}, z_{cg})s \tag{258}$$

Note that

$$G_{cg}(x_{cg}, y_{cg}, z_{cg}) = G^{-1}{}_{cg}(-x_{cg}, -y_{cg}, -z_{cg}) \tag{259}$$

This fact is substantiated by observing Eqs. (253) and (258). Direct multiplication of G_{cg} with $G^{-1}{}_{cg}$ also proves this result by yielding the unit matrix.

Since only the translational inertial velocity components (in cg coordinates) are required for the subsequent analysis, only the first three rows of (258) need to be expanded.

$$u_{cg} = u + z_{cg}q - y_{cg}r \tag{260}$$

$$v_{cg} = v + x_{cg}r - z_{cg}p \tag{261}$$

$$w_{cg} = w + y_{cg}p - x_{cg}q \tag{262}$$

Eqs. (260)-(262) are repeat expressions of Eqs. (214)-(216) in the subroutine FCERQD section.

Since subroutine VELCTY calculates s (and ultimately produces the value of s which fulfills the trim condition), u_{cg}, v_{cg} and w_{cg} can be computed directly, using Eqs. (260)-(262).

Eulerian rotational matrices were discussed in the subroutine EULER section. The transformation matrix required to rotate vectors from cg axes to stability axes can be assembled using $R(\psi)$ and $R(\theta)$ rotations. Normally, the ψ rotation is done first, followed by the $R(\theta)$ rotation. The process is reversed here. Reversing the order of rotation insures that the z_{sa} axis will remain in the x_{cg}-z_{cg} plane of the cg coordinate system. The angles ψ and θ are chosen to fulfill the requirement that the cg's inertial velocity components along the y_{sa} and z_{sa} axes vanish.

The components of cg's inertial translational velocity (in cg coordinates have been denoted u_{cg}, v_{cg}, w_{cg}. When rotated to stability axes, the y_{sa} and z_{sa} components of the inertial velocity vanish. The x_{sa} component thus becomes the inertial speed of the vehicle's cg. Mathematically, this situation is stated as follows:

$$\begin{pmatrix} V_{cg} \\ 0 \\ 0 \end{pmatrix} = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{pmatrix} u_{cg} \\ v_{cg} \\ w_{cg} \end{pmatrix} \tag{263}$$

Inverting and transposing (263) is easily accomplished, noting the characteristics of Eulerian rotational matrices outlined in the section on subroutine EULER.

$$\begin{pmatrix} u_{cg} \\ v_{cg} \\ w_{cg} \end{pmatrix} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} V_{cg} \\ 0 \\ 0 \end{pmatrix} \tag{264}$$

Eq. (264) is easily expanded to solve for u_{cg}, v_{cg}, and w_{cg} in terms of θ, ψ, and V_{cg}:

$$u_{cg} = \cos\theta \cos\psi \; V_{cg} \tag{265}$$

$$v_{cg} = \sin\psi \; V_{cg} \tag{266}$$

$$w_{cg} = - \sin\theta \cos\psi \; V_{cg} \tag{267}$$

Eqs. (265)-(267) are three expressions in the three unknowns ψ, θ, V_{cg}. Of course, u_{cg} v_{cg}, w_{cg} are known (they are computed using Eqs. (260) - (262).

Eq. (266) gives the simple result

$$\sin\psi = \frac{v_{cg}}{V_{cg}} \tag{268}$$

Simple trigonometric manipulation of (268) results in the expression

$$\cos\psi = \frac{\sqrt{V_{cg}^{\,2} - v_{cg}^{\,2}}}{V_{cg}} \tag{269}$$

Combining Eqs. (265) and (269),

$$\cos\theta = \frac{u_{cg}}{\sqrt{V_{cg}^{\,2} - v_{cg}^{\,2}}} \tag{270}$$

Eqs. (267) and (269) combine to yield

$$\sin\theta = - \frac{w_{cg}}{\sqrt{V_{cg}^{\,2} - v_{cg}^{\,2}}} \tag{271}$$

If Eqs.(270) and(271) are squared and added, the following result is obtained:

$$V_{cg}^2 = u_{cg}^2 + v_{cg}^2 + w_{cg}^2 \tag{272}$$

This, of course, is the requirement that must be met, by definition of V_{cg}. Eqs.(268) - (272)can be substituted into Eq. (12). The product of the Ψ and θ Eulerian matrices is the required cg-to-stability axes transformation. Denote this transformation as R:

$$\begin{pmatrix}\text{vector expressed}\\ \text{in stability axis}\\ \text{coordinates}\end{pmatrix} = R \begin{pmatrix}\text{vector expressed}\\ \text{in cg coordinates}\end{pmatrix} \tag{273}$$

R is given in terms of u_{cg}, v_{cg}, and w_{cg} by the following operation. The array shown was generated by multiplying the Eulerian matrices $R(\psi)R(\theta)$, and substituting expressions (268) - (271) for the trigonometric elements that result.

$$R = \begin{bmatrix} \frac{u}{V} & \frac{v}{V} & \frac{w}{V} \\ -\frac{uv}{V\sqrt{V^2 - v^2}} & \frac{\sqrt{V^2 - v^2}}{V} & -\frac{wv}{V\sqrt{V^2 - v^2}} \\ -\frac{w}{\sqrt{V^2 - v^2}} & 0 & \frac{u}{\sqrt{V^2 - v^2}} \end{bmatrix}_{cg} \tag{274}$$

(the subscript cg has been omitted from symbols within the matrix for simplicity in notation)

V_{cg} is given by(272). Note that R becomes indeterminate if $V_{cg} \rightarrow 0$. This is to be expected, since stability axes are undefined when the aircraft cg has no inertial speed. Note also that the division $\sqrt{V_{cg}^2 - v_{cg}^2}$ can never be zero (except in the indeterminate case when $V = 0$).

The matrix R is used to rotate three-element columns (vectors) from cg to stability axes. In order to facilitate the resolution of Δp_c and Δs_{cg} (which are six-element columns made up of three-element subcolumns), define the expanded rotational matrix R.

$$R = \begin{bmatrix} R & 0 \\ 0 & R \end{bmatrix} \tag{275}$$

Eq. (256) can be premultiplied by $\bar{R}$ to yield an expression for overall vehicle loading in stability axis system coordinates. Also, since $R^{-1}R = R^TR$ = the unit matrix, the product R^TR can be inserted into (256) just in front of columns Δs_{cg} and $\Delta \dot{s}_{cg}$.

$$\Delta p_{sa} = R \left[G_{cg}{}^T P_s G_{cg} \right] R^T \Delta s_{sa} + R \left[G_{cg}{}^T P_{\dot{s}} G_{cg} \right] R^T \Delta \dot{s}_{sa}$$

$$+ R \left[G_{cg}{}^T P_c \right] \Delta C \tag{276}$$

Eq. (276) derives from (256) in this manner because

$$\Delta s_{sa} = R \; \Delta s_{cg} \tag{277}$$

$$\Delta \dot{s}_{sa} = R \; \Delta \dot{s}_{cg} \text{ , and} \tag{278}$$

$$\Delta p_{sa} = R \, \Delta p_{cg} \tag{279}$$

To simplify the notation, define the 6 x 6 matrix X:

$$X = R \, G_{cg}{}^T \tag{280}$$

Noting that $X^T = G_{cg} R^T$, Eq. (276) is written

$$\Delta p_{sa} = \left[X \, P_s \, X^T \right] \Delta s_{sa} + \left[X \, P_{\dot{s}} \, X^T \right] \Delta \dot{s}_{sa}$$

$$+ \left[X \, P_c \right] \Delta c \tag{281}$$

The matrices in brackets in Eq.(281) are the stability derivative arrays with respect to stability axes. Eqs. (251), (257) (260) - (262), (274), (275), (280), and (281) are programmed in MOSTAB, so that the stability derivative arrays can be expressed with respect to aircraft axes, cg axes and stability axes.

The R transformation (Eq.274) can be used to convert the vehicle inertia tensor, I, expressed in overall vehicle coordinates, to the inertia tensor expressed in stability axes.

$$I_{sa} = R\, I R^T \tag{282}$$

where

$$I \triangleq \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{zz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix}$$

Eq.(282) is derived using arguments identical to those used in deriving (276).

III. ROTOR ANALYSIS

III.1 INTRODUCTION

The rotor analysis is presented in this part in relatively general form. The simplified version of the analysis presently used in MOSTAB is assembled in the main body of this work, referring to the general analysis given here for the basic equations. If it becomes necessary to remove some of the present MOSTAB simplifications, the equations developed here can be used to add the desired effects with no substantial amount of additional analytic work.

Rotor types which do not couple loads among the blades (except through rotor shaft motion and aerodynamic interference) are addressed in this part. Teetering rotors (floating hub rotors) and rotors with independently articulated blades with coupling links or cables do not generally fall into this category, because these rotors couple loads among the blades without first applying such loads to the shaft. To account for such coupling, certain terms must be added to the blade motion equations. No difficulty in extending the present analysis to include direct load coupling is anticipated. Usually, load-coupled rotors can be approximated with independently articulated rotor models for vehicle handling quality, and stability examinations. The coupling among blades influences vibration levels, blade stresses, etc., but usually has negligible effect on aircraft handling characteristics.

III.2 AXIS SYSTEM DEFINITION

The reference point for all rotors lies at the intersection of the shaft centerline and the unflexed blades' quarter chord line (see Part I for definition of "reference points". If such a point is undefined because of curved blades or some other geometric difficulty, the point lies at a convenient point in the rotor hub, on the shaft centerline.

The rotor's local axis system is fixed to the nonrotating airframe with its z axis coincident with the rotor shaft centerline and its origin at the rotor's reference point. The azimuthal position of this system is defined (with respect to the nonrotating airframe) in any convenient manner. Constant Euler angles $\psi_r, \theta_r, \varphi_r$ are defined which locate these local rotor hub axes with respect to overall vehicle reference axes. This local axis system will henceforth be referred to as the rotor's "hub axes."

Now define the "rotor axes" as a system that rotates with the rotor hub. Then the rotor axes have an angular speed Ω with respect to the hub axes. The z axis of the rotor axes lies coincident with the z hub axis, and the origins of the two systems are coincident. Define ψ as the azimuthal angle between the rotor and hub axes. When $\psi = 0$, the rotor and hub axes are coincident. Clearly, when $\dot{\Omega} = 0$,

$$\psi = \Omega t + \psi_0 \tag{283}$$

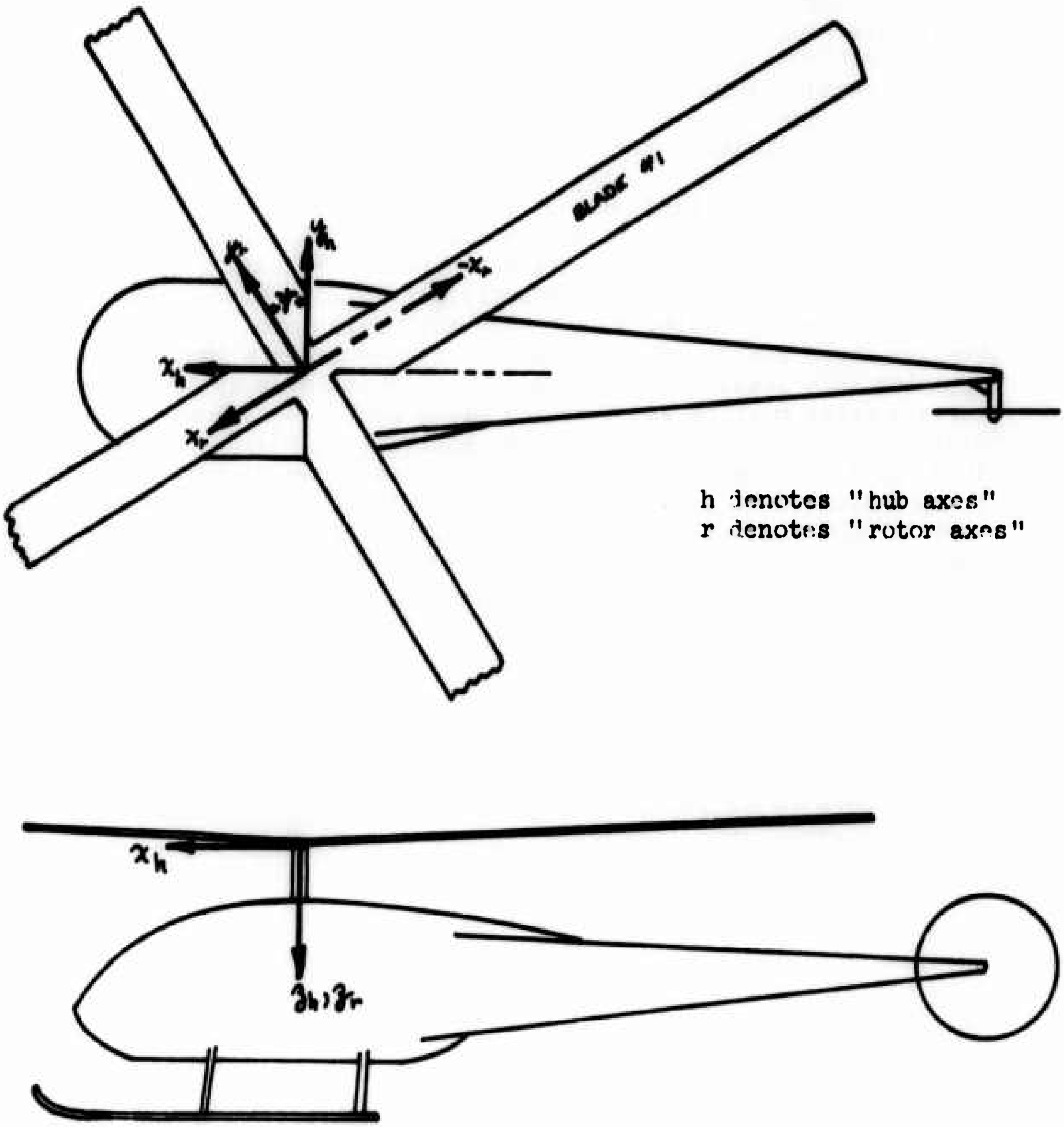

Figure 11. Rotor Axis Systems.

For most U. S.-built helicopters, the azimuthal angle ψ implies a negative Euler azimuthal (z axis) rotation. Care must be exercised to define the sign of Ω correctly when using MOSTAB on a given rotor. A single vehicle may have rotors which have not only varying values of Ω, but varying signs as well.

Figure 11 shows a conventional helicopter, with the "hub" and "rotor" axes illustrated as they apply to the main rotor. A similar pair of axes apply to the tail rotor, but these are not shown by Figure 11. Note that the equation for ψ above includes an "initial" constant, ψ_o. ψ_o is chosen so that the shaft normal plane projection of rotor blade number 1 lies along the -x rotor axis. With this definition of ψ_o, the azimuthal angle ψ used here is the conventional angle used in most classical rotor analyses (particularly the bulk of the work published by NACA).

III.3 BLADE REFERENCE LINE

Define a blade reference line (BRL) along the span of blade number 1. This reference line is attached to the mass molecules of the structure. Its exact position on the blade is arbitrary, but the quarter chord line is probably the most convenient choice.

Figure 12 shows the BRL and the "rotor axes." The reference line intersects the rotor axis system origin. In the analysis that follows, the BRL is assumed to be infinitely stiff in tension (it cannot stretch).

The coordinate, s, defines some point, P, on the BRL. Regardless of the shape of the BRL, s defines a particular mass element of the blade. Thus, s is a measurement of the length of the BRL segment between P and the rotor axis system origin. s is constant for a given P because of the assumption that the BRL cannot stretch.

Figure 12. Blade Reference Line.

When the BRL is deformed, it generally will have coordinates x(s), y(s), z(s) which define its shape in rotor coordinates. With the assumption that the line does not stretch,

however, the coordinate $x(s)$ can be expressed as a function of s, $y(s)$ and $z(s)$.

Distributed loading functions are applied to the BRL from two sources:

(a) "Apparent" loading due to acceleration of the blade mass in inertial space (which can include gravity forces if desired)

(b) Aerodynamic loading.

This loading picture can be expressed as a distributed force vector, $\bar{F}$, expressed in rotor coordinates:

$$\bar{F}(s,t)=\hat{i}\left[p_{xi}(s,t)+p_{xa}(s,t)\right]+\hat{j}\left[p_{yi}(s,t)+p_{ya}(s,t)\right]+\hat{k}\left[p_{zi}(s,t)+p_{za}(s,t)\right]$$

The distributed loading function expressed above can be integrated with respect to s from blade root to tip, resulting in a time varying expression for rotor shaft forces due to one blade. Integrating $\bar{r}\times\bar{F}$ produces the shaft moments.* These integrations are considered in detail in a later section of this appendix.

III.4 INERTIAL ANALYSIS

The inertial analysis presented here is required to generate expressions for the "apparent" inertial loading of a rotor blade as it accelerates in inertial space. The results of the analysis will be expressions for the distributed loading functions denoted p_{xi}, p_{yi} p_{zi}. These loading components are expressed in rotor coordinates.

Consider Figure 12 which shows a portion of blade ds long, at the point s on the BRL. The mass of this piece of blade is given by the expression

$$dM = m(s)\ ds \tag{284}$$

where $m(s)$ is the blade mass distribution.

Figure 13 shows dM with associated vectors to be used for the subsequent inertial analysis. The rotor axes are shown as they relate to the "hub" axes. As discussed in the main body of this report, the hub axis system is defined with its origin coincident with the rotor system origin and its z_h axis coincident with the z rotor axis. Generally, the hub axes will be fixed rigidly to a flight vehicle. As far as the present problem is concerned, however, the hub axes are defined as above, with the further stipulation that the motion of x_h, y_h, z_h in space is

* The position vector $\bar{r}$ is defined in rotor coordinates as

$$\bar{r} = \hat{i}\ x + \hat{j}y + \hat{k}z$$

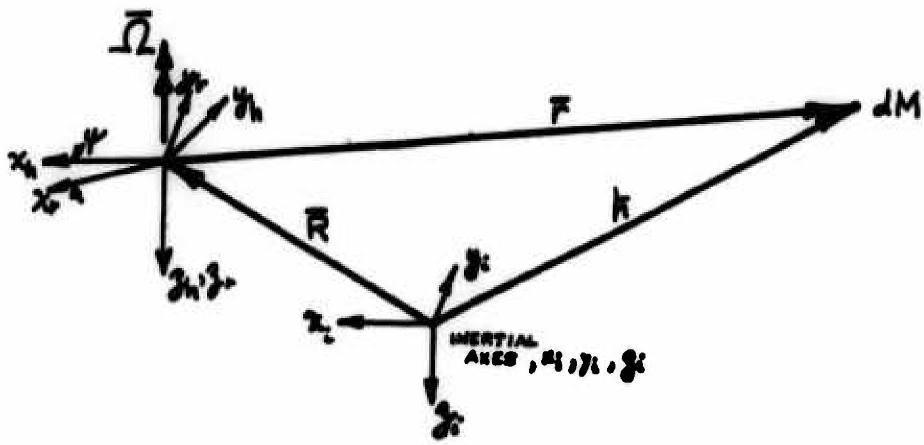

Figure 13. Vectors for Inertial Analyses.

given, and that the angle ψ (in Figure 13) is defined as a function of time.

The vector, Ω, in Figure 13 is the rotor speed, and is defined

$$\bar{\Omega} = -\hat{k}\frac{d\psi}{dt} = -\hat{k}_h\frac{d\psi}{dt} \tag{285}$$

where $\hat{i}$, $\hat{j}$, $\hat{k}$ are rotor axis system unit vectors and $\hat{i}_h$, $\hat{j}_h$, $\hat{k}_h$ are hub axis system unit vectors.

Newton's second law expresses the force on dM as

$$d\bar{F}_i = dM\left(\frac{d^{*2}\bar{h}}{dt^2}\right) \tag{286}$$

where the asterisk on the differentiation symbol indicates differentiation in the inertial axis system. From inspection of Figure 13, Eq.(286) becomes

$$d\bar{F}_i = dM\left(\frac{d^{*2}\bar{R}}{dt^2} + \frac{d^{*2}\bar{r}}{dt^2}\right) \tag{287}$$

The quantity $\frac{d^{*2}\bar{R}}{dt^2}$ is assumed a known function to this problem.

It will be available in components g_x, g_y, g_z in hub axis coordinates (x_h, y_h, and z_h in Figure 13). The z_h rotation between the hub and rotor axes is $-\psi$. Then the components of $\frac{d^{*2}\bar{R}}{dt^2}$ in rotor axes, from inspection of Figure 13, are

$$\frac{d^{*2}R}{dt^2} = i\,(g_x \cos\psi - g_y \sin\psi) + j\,(g_x \sin\psi + y_y \cos\psi) + kg_z \tag{288}$$

The Coriolis theorem is derived in most works on classical mechanics (e.g. Reference 4). Written as applicable to Figure 13, this theorem becomes

$$\frac{d^2 r}{dt^2} = \ddot{r} + \bar{\omega} \times (\omega \times r) + 2\,\omega \times \dot{r} + \dot{\bar{\omega}} \times r \tag{289}$$

where the "dot" denotes differentiation with respect to time in rotor axes. The variable, $\bar{\omega}$, is the "spin" rate of rotor axes with respect to inertial axes. In conventional airplane notation, the spin rate of hub axes would have components p, q, r. Since $-\hat{k}_h\Omega$ is the rotor spin rate with respect to the hub axes, the spin rate of the rotor axes, ω, is given by the expression

$$\bar{\omega} = \hat{i}_h p + \hat{j}_h q + \hat{k}_h (r - \Omega) \tag{290}$$

The vector ω can be expressed in rotor system coordinates by resolution through the angle ψ. The result is

$$\omega = \hat{i}\,(p \cos\psi - q \sin\psi) + \hat{j}\,(p \sin\psi + q \cos\psi) + \hat{k}\,(r - \Omega) \tag{291}$$

From Figure 12, one sees that $\bar{r}$ can be expressed in rotor coordinates as

$$r = \hat{i}x + \hat{j}y + \hat{k}z \tag{292}$$

Eqs.(288), (289), (291), and (292) can be processed by methods of ordinary vector calculus to produce an expanded expression for $\dfrac{d^{*2}h}{dt^2}$ in rotor coordinates. Knowing this vector quantity, the differential force d F_i of Eq.(287) can be determined. Note that dF_i is the force on blade mass dM, applied by the blade structure in order to produce $\dfrac{d^{*2}h}{dt^2}$. Taking the D' Alembert approach of viewing mass accelerations as apparent forces, the force on the blade structure applied by the accelerating blade mass is $-dF_i$. The components of $-dF_i/ds$ are the "inertial" distributed load functions p_{xi}, p_{yi}, and p_{zi}. Expressed

in expanded form, these functions are

$$P_{xi}(s,t) = - m(s) \Big| g_x \cos\psi - g_y \sin\psi + \ddot{x} - x(r-\Omega)^2$$

$$+ rz(p\cos\psi - q\sin\psi) - 2\dot{y}(r-\Omega) + z(\dot{p}\sin\psi + \dot{q}\cos\psi)$$

$$- y(\dot{r}-\dot{\Omega}) + (p\sin\psi + q\cos\psi)\left[2\dot{z} + y(p\cos\psi - q\sin\psi)\right.$$

$$\left. - x(p\sin\psi + q\cos\psi)\right]\Big| \tag{293}$$

$$p_{yi}(s,t) = -m(s) \Big| g_x \sin\psi + g_y \cos\psi + \ddot{y} - y(r-\Omega)^2 + rz(p\sin\psi$$

$$+ q\cos\psi) + 2\dot{x}(r-\Omega) - z(\dot{p}\cos\psi - \dot{q}\sin\psi) + x(\dot{r}-\dot{\Omega})$$

$$- (p\cos\psi - q\sin\psi)\left[2\dot{z} + y(p\cos\psi - q\sin\psi)\right.$$

$$\left. - x(p\sin\psi + q\cos\psi)\right]\Big| \tag{294}$$

$$p_{zi}(s,t) = - m(s) \Big| g_z + \ddot{z} - x\left[(\dot{p}-2\Omega q)\sin\psi + (\dot{q} + 2\Omega p)\cos\psi\right]$$

$$+ rx(p\cos\psi - q\sin\psi) - 2\dot{x}(p\sin\psi + q\cos\psi)$$

$$+ y\left[(\dot{p} - 2\Omega q)\cos\psi - (\dot{q} + 2\Omega p)\sin\psi\right] + ry(p\sin\psi$$

$$+ q\cos\psi) + 2\dot{y}(p\cos\psi - q\sin\psi) - z(p^2 + q^2)\Big| \tag{295}$$

III.5 AERODYNAMIC ANALYSIS

This analysis is required to produce expressions for the distributed aerodynamic blade loading functions p_{xa}, p_{ya}, p_{za}. These distributed loading functions are vector components of the aerodynamic loading referenced to rotor axes.

The aerodynamic loading at a point on the BRL depends upon the velocity of air with respect to the blade at that point. To derive an expression for this velocity, consider Figure 13. Instead of space axes, envision the axis system attached to the air mass in the vicinity of the rotor hub. Then the velocity of the point s on the BRL, with respect to the local air mass, can be expressed as

$$\frac{d^*\bar{h}}{dt} = \frac{d^*\bar{R}}{dt} + \frac{d^*\bar{r}}{dt} \tag{296}$$

The quantity $\frac{d^*\bar{R}}{dt}$ represents the translational airspeed of the hub axes (or rotor axes), and is given to this problem. If the components of velocity of the rotor hub axes with respect to the local air body are denoted u_a, v_a, w_a, then

$$\frac{d^*\bar{R}}{dt} \triangleq \hat{i}_h u_A + \hat{j}_h v_A + \hat{k}_h w_A \tag{297}$$

Resolving this velocity into rotor coordinates,

$$\frac{d^*\bar{R}}{dt} \triangleq \hat{i} (u_A \cos\psi - v_A \sin\psi) + \hat{j} (u_A \sin\psi + v_A \cos\psi) + \hat{k} w_A \tag{298}$$

The rotational airspeed of the hub axes is also given to this problem. Denote this airspeed $\bar{\omega}_{hA}$:

$$\bar{\omega}_{hA} \triangleq \hat{i}_h p_A + \hat{j}_h q_A + \hat{k}_h r_A \tag{299}$$

Eq. (299) resolves to rotor axes in exactly the same way Eq. (297) did, with p_a, q_a, r_a, respectively substituted for u_a, v_a, w_a. Noting that the rotational speed of rotor axes with respect to hub axes is $-\hat{k}\Omega$, one writes the rotational airspeed of the rotor axes as

$$\bar{\omega}_A = \hat{i} (p_A \cos\psi - q_A \sin\psi) + \hat{j} (p_A \sin\psi + q_A \cos\psi) + \hat{k} (r_A - \Omega)$$

Classical works on vector mechanics (Reference 4) demonstrate that

$$\frac{d^*\bar{r}}{dt} = \dot{\bar{r}} + \omega_A \times \bar{r} \tag{300}$$

where again, the dot denotes differentiation with respect to time in rotor axes. The quantities $\dot{\bar{r}}$ and $\bar{\omega}_A \times \bar{r}$ are easily calculated. Combining the result of this calculation with Eqs. (296) and (297),

$$\bar{V}_A \triangleq \frac{d^*\bar{h}}{dt} = \hat{i}\left[u_A \cos\psi - v_A \sin\psi + \dot{x} + z\,(p_A \sin\psi + q_A \cos\psi) - y\,(r_A - \Omega)\right]$$
$$+ \hat{j}\left[u_A \sin\psi + v_A \cos\psi + \dot{y} + x\,(r_A - \Omega) - z\,(p_A \cos\psi - q_A \sin\psi)\right]$$
$$+ \hat{k}\left[w_A + \dot{z} + y\,(p_A \cos\psi - q_A \sin\psi) - x\,(p_A \sin\psi + q_A \cos\psi)\right] \quad (302)$$

Eq. (302) gives the velocity of a point, s, on the BRL (noting that s specifies x(s), y(s), z(s), ż(s), etc) with respect to the local air body.

Since airloads are generated by an airfoil section at s, the velocity $\frac{d^*h}{dt}$ must be expressed in coordinates directly associated with the airfoil section, instead of rotor system coordinates. To do this, first define the "blade" axis system at s as follows:

(a) The origin of the blade axes lies at s (point P of Figure 12).

(b) The x_b axis is tangent to the BRL at s, and points generally toward the rotor hub.

(c) The y_b axis lies parallel to the chord of the blade section at s.

Define three angles, ν, τ, ζ, to represent the three Eulerian coordinate rotations required to rotate the rotor axis system to a position parallel to the blade system. Then a transformation matrix T can be assembled, such that

$$\begin{pmatrix}\text{vector expressed}\\ \text{in blade}\\ \text{coordinates}\end{pmatrix} = T\,(\nu, \tau, \zeta) \begin{pmatrix}\text{vector expressed}\\ \text{in rotor}\\ \text{coordinates}\end{pmatrix} \qquad (303)$$

where

$$T \triangleq \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\zeta & \sin\zeta \\ 0 & -\sin\zeta & \cos\zeta \end{bmatrix} \begin{bmatrix} \cos\tau & 0 & -\sin\tau \\ 0 & 1 & 0 \\ \sin\tau & 0 & \cos\tau \end{bmatrix} \begin{bmatrix} \cos\nu & \sin\nu & 0 \\ -\sin\nu & \cos\nu & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\tau\ \cos\nu & \cos\tau\ \sin\nu & -\sin\tau \\ \sin\zeta\ \sin\tau\ \cos\nu\ -\cos\zeta\ \sin\nu & \sin\zeta\ \sin\tau\ \sin\nu\ +\cos\zeta\ \cos\nu & \sin\zeta\ \cos\tau \\ \cos\zeta\ \sin\tau\ \cos\nu\ +\sin\zeta\ \sin\nu & \cos\zeta\ \sin\tau\ \sin\nu\ -\sin\zeta\ \cos\nu & \cos\zeta\ \cos\tau \end{bmatrix} \tag{304}$$

The angles ν and τ as applied to a rotor BRL will generally be small, representing the blade flapping and hunting angles respectively. The angle ζ will be somewhat larger, being approximately equal to the blade feathering angle with respect to the rotor shaft normal plane. The geometric relationships among the variables τ, ν and ζ in T, and the blade reference line coordinates x(s,t), y(s,t) and z(s,t) will be examined in Section I. The remainder of the aerodynamic analysis requires only the definition of the rotor-to-blade axis system transformation Eq. (303).

If u_s, u_c and u_n denote the spanwise, chordwise and normal-to-chordwise airspeeds at the airfoil section at s, then the T transformation matrix defined by (303) can be used as follows:

$$\begin{pmatrix} u_s \\ u_c \\ u_n \end{pmatrix} = T\ V_A \tag{305}$$

where $\bar{V}_A$ is the airspeed at s in rotor coordinates defined by Eq. (302).

In general, the aerodynamic forces on a blade element will be complicated, nonlinear functions of u_n and u_c, and functions of the characteristics of the airfoil section being considered. Here, say functions f_n and f_c are available, such that

$$f_n = f_n(u_n, u_c, s) \tag{306}$$

$$f_c = f_c(u_n, u_c, s) \tag{307}$$

where f_n is the distributed aerodynamic force normal to the blade section chordline and f_c is the distributed aerodynamic force parallel to the blade section chordline.

Figure 14 shows f_n, f_c, u_n and u_c with the airfoil section at s. The section can be viewed as the facing end of a blade element of length ds. Such an element develops the air forces as shown by the diagram.

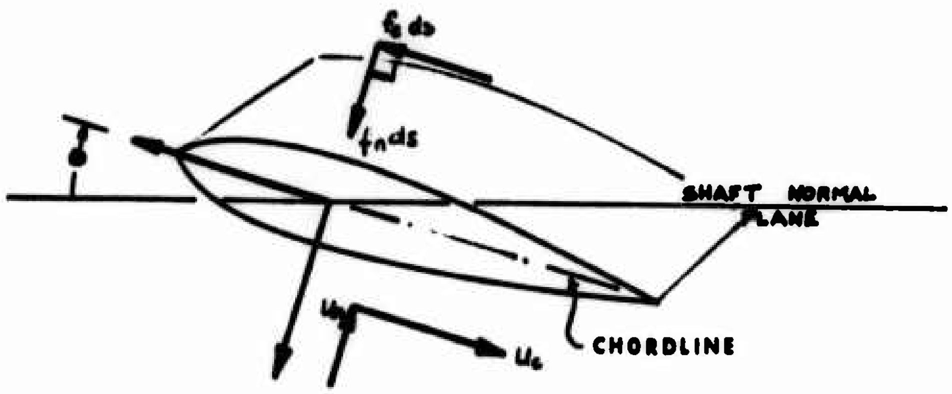

Figure 14. Aerodynamic Forces on Blade Element.

In the diagram, u_n and u_c depict the velocity components of the air with respect to the section.

The transformation matrix T can be used to determine blade aerodynamic distributed loading functions p_{xa}, p_{ya} and p_{za} in rotor coordinates. Taking advantage of the fact that, since T is an Eulerian transformation matrix, $T^{-1} = T^T$ (T (inverse) = T (transpose)),

$$\begin{pmatrix} p_{xa} \\ p_{ya} \\ p_{za} \end{pmatrix} = T^T \begin{pmatrix} 0 \\ f_c \\ f_n \end{pmatrix} \tag{308}$$

The equations developed in this section are sufficient to define p_{xa}, p_{ya} and p_{za}, given ψ, u_A, v_A, w_A, p_A, q_A, r_A, x, y, z, ν, τ, ζ, Ω and suitable aerodynamic functions (306) and (307).

III.6 SIMPLIFIED AERODYNAMICS

A linear lift coefficient function and a simple parabolic drag polar can be used in lieu of complex aero functions for f_n and f_c. This simplified approach requires small angle-of-attack approximations.

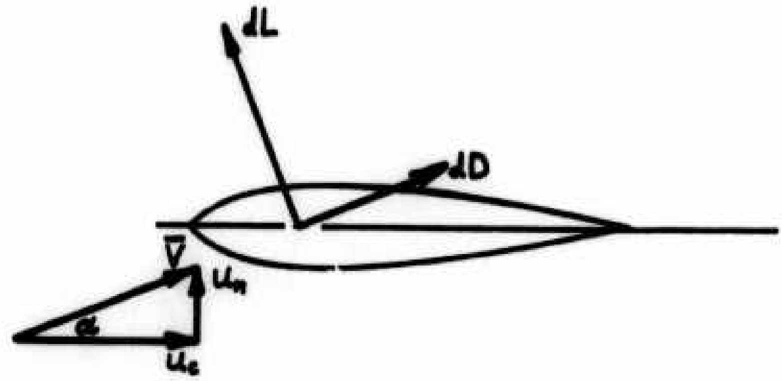

Figure 15. Blade Element Aerodynamics for Small Angle-of-Attack.

The airfoil section of Figure 15 has infinitesimal span ds. It develops infinitesimal forces dL and dD.

For small α,

$$\alpha = \frac{u_n}{u_c} \tag{309}$$

$$\bar{V} = u_c \tag{310}$$

Assume a linear lift curve slope a, and a simple parabolic drag polar:

$$C_L = a\alpha = a\left(\frac{u_n}{u_c}\right) \tag{311}$$

$$C_D = \delta_o + \delta_1\alpha + \delta_2\alpha^2 = \delta_o + \delta_1\left(\frac{u_n}{u_c}\right) + \delta_2\left(\frac{u_n}{u_c}\right)^2 \tag{312}$$

The distributed lift and drag forces are

$$\frac{dL}{ds} = L' = \left(\frac{au_n}{u_c}\right)\frac{c}{2}\,uc^2 = a\left(\frac{\rho c}{2}\right)u_n u_c \tag{313}$$

$$\frac{dD}{ds} = D' = \left(\delta_o + \delta_1\left(\frac{u_n}{u_c}\right) + \delta_2\frac{u_n^2}{u_c^2}\right)\frac{\rho c}{2}\,u_c^2$$

$$= \frac{\rho c}{2}\left(\delta_o u_c^2 + \delta_1 u_n u_c + \delta_2 u_n^2\right) \tag{314}$$

Resolving these distributed forces to lie normal and parallel to the chord line

$$f_n = -L'\cos\alpha - D'\sin\alpha = -a\left(\frac{\rho c}{2}\right)u_n u_c - \frac{\rho c}{2}\left(\delta_o u_n u_c\right) - \frac{\rho c}{2}\,\delta_1 u_n^2$$

$$= -\left(\frac{\rho c}{2}\right)\left[\left(a + \delta_o\right)u_n u_c + \delta_1 u_n^2\right] \tag{315}$$

$$f_c = L'\sin\alpha - D'\cos\alpha = a\left(\frac{\rho c}{2}\right)u_n^2 - \left(\frac{\rho c}{2}\right)\left(\delta_o u_c^2 + \delta_1 u_n u_c + \delta_2 u_n^2\right)$$

$$= -\left(\frac{\rho c}{2}\right)\left[\left(\delta_2 - a\right)u_n^2 + \delta_o u_c^2 + \delta_1 u_n u_c\right] \tag{316}$$

where the term $\frac{u_n^3}{u_c}$ has been neglected.

Eqs.(315) and(316) provide functions of the form (306) and (307) assuming simplified models for the aerodynamic lift and drag coefficients.

III.7 TOTAL LOADS ON THE BRL

The total loading on the blade comes from the summation of inertial and aerodynamic forces. The components of this loading, in rotor axis coordinates,

$$p_x\ (s,\ t) = p_{xa} + p_{xi} \tag{317}$$

$$p_y\ (s,\ t) = p_{ya} + p_{yi} \tag{318}$$

$$p_z\ (s,\ t) = p_{za} + p_{zi} \tag{319}$$

III.8 BLADE MOTION EQUATIONS - THE NORMAL MODE METHOD

The preceding sections of this part show derivations of the expressions for distributed loading on aerodynamic rotor blades. If the blades are rigid (as can usually be assumed for most propellers), the BRL coordinates in these expressions are only functions of s. Since the time varying quality is removed from these coordinates due to the rigid blade constraint, rotor geometry defines the functions x(s), y(s), z(s) and θ(s), and no further analysis is required to determine these functions. In this case, the distributed loading functions can be defined as soon as the aerodynamic and inertial motions of the rotor hub-axes are known.

If the blades on an aerodynamic rotor are flexible (as in the case of hinged blades, or so called "rigid" rotors whose blades deform elastically to a significant degree), the BRL coordinates are functions of time, as well as the spatial coordinate s. Determination of these coordinates as functions of time (and, of course, s) might be called the blade motion problem.

References (3) and (5) show the application of the normal method to the problem of thin flexible beams moving under external, time varying distributed loading functions. Both references address the one-dimensional beam motion problem, but the method is readily extendable to the multidimensional problem. Discussion of such extension is deferred to a later section of this part.

The salient features of the one-dimensional thin beam motion problem are presented now, in the notation of Reference 5. Figure 16 below shows the coordinates

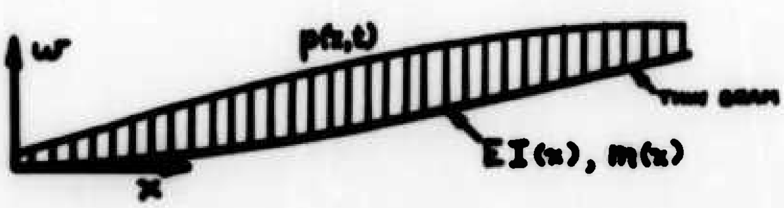

Figure 16. Coordinates for Blade Motion Analysis.

The problem is to specify w(x,t), given the structural and inertial properties of the beam, and the external function p(x,t).

If $p(x,t)=0$, the beam will vibrate if disturbed from rest. Many methods are available for analyzing this free vibration problem. The results of these analyses show that the beam has an infinite number of "normal modes" of vibration, each occurring at a different frequency. These frequency values are usually called the natural frequencies of vibration, or the eigenvalues of the flexible beam problem. The eigenvalues are functions of the beam's stiffness distribution $EI(x)$, mass distribution $m(x)$, and supports (simply supported, cantilevered, etc.). A function, $\Phi_i(x)$, is associated with each natural mode of vibration. This function is called the modeshape or eigenfunction of the i'th normal mode of vibration. If the beam is vibrating in its j'th normal mode only, then the coordinate $w(x,t)$ is given as

$$w(x,t) = K\, \Phi_j(x) \sin(\omega_j t + \theta) \tag{320}$$

where ω_j is the eigenvalue associated with the j'th normal mode, and K and θ are arbitrary constants determined by the initial disturbance which started the motion.

The normal mode method is essentially a functional series solution method for the forced beam. The coordinate $w(x,t)$ is expressed as a series in the modeshapes, and a set of "generalized" or "normal" coordinates:

$$w(x,t) = \sum_{i=1}^{n} \Phi_i(x) \eta_i(t) \tag{321}$$

In the case of the continuous beam, n is infinite. The generalized coordinate $\eta_i(t)$ specifies how much of the i'th eigenfunction is involved in the shape of the beam at time t.*

References 3 and 5 show that the eigenfunctions are orthogonal with respect to the beam mass distribution, $m(x)$. This very important property is expressed by the equation

$$\int_0^L m(x)\, \Phi_i(x) \Phi_j(x)\, dx = 0 \text{ when } i \neq j \tag{322}$$

This condition of orthogonality accounts for the name "normal" as applied to the vibration problem.

---

* Observe that Eq.(320) applies only when the beam is vibrating in its j'th mode with no external disturbances. Eq.(321) is the assumed functional series solution for the beam as it moves under any specified external excitation.

Now consider the case when $p(x,t)$ is nonzero. A partial differential equation can be written in independent coordinates x (space) and t (time). The functional series expression (320) is applied to the equation as a change of coordinates. Essentially, the normal coordinates, $\eta_i(t)$, are substituted for the distributed coordinate $w(x,t)$. Using the condition of orthogonality expressed by Eq.(322), the infinite number of equations in the infinite number of coordinates $\eta_i(t)$ are decoupled. The form of the decoupled equations is

$$\ddot{\eta}_r + \omega_r^2 \eta_r = \frac{N_r}{M_r} \qquad r = 1,2 \cdots\cdots \infty \tag{323*}$$

where η_r, M_r, and N_r are the normal coordinate, "generalized mass" and "generalized force" (respectively) associated with the r'th normal mode. The generalized mass is constant, and given by the expression

$$M_r = \int_0^L m(x)\Phi_r^2(x)dx \tag{324}$$

The generalized force is given as

$$N_r = \int_0^L \Phi_r(x)\ p\ (x,t)\ dx \tag{325}$$

The normal mode method for calculating the motion of forced beams has been presented above, as it applies to one-dimensional motion. The method is in no way restricted to one-dimensional problems, and is easily extended to the n-dimensional forced motion problem. Since MOSTAB considers only blade flapping motion at the present time, the multicoordinate extension of the modal method is not presented here. If necessary, the extension can be made to include inplane and torsional blade modes. The number of normal coordinate is increased if this is done, but no theoretical or practical difficulty restricts MOSTAB to the simple one-dimensional case presently incorporated. Reference 3 shows how the normal mode method can be used to study flutter. In an equivalent manner, MOSTAB could be extended (with the accompanying increase in computer time requirements) to even include blade flutter modes.

* The damping term shown in Reference 5 is dropped here. This issue will be considered later.

III.9 GENERAL COMMENTS ON THE NORMAL MODE METHOD

In any practical solution, a finite number of normal coordinates, $\eta_i(t)$, must be selected to represent flexible body motion. Usually, the coordinates associated with the lowest frequencies of the elastic system are chosen. The normal mode method is very practical for aerodynamic rotor blades, because a minimal number of degrees-of-freedom (normal coordinates) need be chosen to represent the significant characteristics of the blades. (This is particularly true when overall vehicle handling qualities and stability are being considered). Experience has shown that the influence of blade motion on handling qualities is represented by the first flapping mode only. For some stability augmentation schemes, the first inplane (or first "chord") mode may also influence the stability and handling qualities of an aircraft. Higher frequency blade modes are important with regard to blade stability (flutter), rotor stability and vibration considerations (including structural fatigue problems), but these modes seldom affect vehicle dynamics.

Sometimes static torsional deflections of blades influence vehicle dynamics. This is not due to a torsional vibration mode (since it is a <u>static</u> effect), so consideration of this phenomenon does <u>not</u> require the addition of another blade degree of freedom. This issue is mentioned here to substantiate the claim made above that higher order <u>modes</u> (including the blade's first torsion mode) seldom influence vehicle dynamics.

The normal mode method has many advantages when used to study vehicle dynamics. The following considerations are applicable in this regard:

(a) The structural characteristics of the blades, and the influence of centrifugal force, are represented in the modeshape functions, $\phi_i(x)$, and frequencies, ω_i. It is easy to estimate modeshapes and frequencies, if detailed structural information is unavailable for the given blade. For hinged blades, the first modeshape and frequency (for both chord and flapping motion) can be determined from blade geometry. Reference 6 gives an extensive tabulation of modeshapes and frequencies for pinned and cantilever beams (rotating and nonrotating) with linearly varying structural characteristics, and with tip weights. These charts can be used to get good estimates of modeshapes and frequencies, if detailed data is unavailable.

(b) The dynamics of a vehicle are usually quite insensitive to errors in estimated modeshape functions.

Very primitive models can be used for these functions, while still achieving accurate results. It is only important that the modeshape function be <u>compatible</u> with the mass distribution estimated. When estimating a modeshape function, use the mass distribution which will produce this shape. If Reference 6 is used to estimate modeshapes, this requirement is automatically

(c) Usually, very stringent rotor design criteria restrict the frequencies that a blade must possess. This is particularly true for the lower blade frequencies. Knowledge of these requirements makes it quite easy to estimate frequencies, even for undesigned rotors being studied for predesign evaluation.

Some concern must be given to the function $p(x,t)$ when the normal mode method is applied to aerodynamic rotors. Eq. (331) gives an expression for the generalized forcing function in terms of the distributed force. In the case of aerodynamic rotors, $p(x,t)$ will contain all aerodynamic forcing effects, and most inertial forces. Certain terms in the inertial distributed loading functions refer to loads caused by BRL acceleration or position in the rotor axis system. The terms $-m\ddot{y}$, $+ my(r-\Omega)^2$ in Eq. (294), and $-m\ddot{z}$ in Eq. (295) are in this category. These terms are included in the equations used to perform the vibration analysis, leading to the determination of the eigenvalues, ω_i, and eigenfunctions, Φ_i, for the "unloaded" BRL. The influence of blade tension will also be included in the vibration analysis. Since these terms are included in the modeshapes and frequencies, they must be excluded from the distributed functions that are integrated to get the generalized forces.

Considerable flexibility exists as to which inertial (and possibly aerodynamic) forcing terms are to be included in the vibration analysis (and thus excluded from the generalized forces). Reference 5 shows a constant coefficient damping term on the left side of Eq. (323). (Eq. (323) implies that the damping is included in the generalized force term.) When the modal analysis is applied to an aerodynamic rotor, the generalized force terms usually contain effective spring, damping and mass terms (i.e., the influence of η_r, $\dot{\eta}_r$ and $\ddot{\eta}_r$ is present in the function N_r of Eq. (323). These terms usually appear with time varying coefficients or in nonlinear formulations, which makes it necessary to include them in the generalized force term instead of on the left side of the equation. Experience has shown that the damping terms can be included in the generalized forces as Eq. (323) is integrated numerically, with no observable errors. It is desirable to include as many spring and mass terms in the

vibration analysis as practicable, however, excluding these terms from the generalized forces. (Of course this cannot be done if the terms are nonlinear or time varying). This is particularly true if very low levels of damping are available for one or more of the modes (e.g., the inplane mode on rotors with no inplane dampers).

III.10 CALCULATING THE BLADE REFERENCE LINE ANGLES AND COORDINATES

The previous sections of this part have presented the basic normal mode method for calculating blade motions. The solutions of the modal equations, in conjunction with the assumed modeshapes, eventually yield expressions for the blade reference line coordinates y(s,t) and z(s,t). As can be seen from inspection of the inertial and aerodynamic loading expressions, the coordinate x(s,t) is also required.

The transformation (303) can be used to derive the necessary expression for x(s,t), and to relate the coordinates ν and τ to y(s,t) and z(s,t). Consider the differential length of blade reference line, ds, as depicted by Figure 12. Since the blade axes have their x_b axis tangent to the BRL at s, the increment ds can be considered a vector pointed in the $-x_b$ direction. The vector $-\hat{i}_b\,ds$ will have components $\hat{i}\,dx + \hat{j}\,dy + \hat{k}\,dz$ in rotor axes, which are easily determined from Eq. (303).

$$\begin{pmatrix} -ds \\ 0 \\ 0 \end{pmatrix} = T \begin{pmatrix} dx \\ dy \\ dz \end{pmatrix} \tag{326}$$

or

$$\begin{pmatrix} dx \\ dy \\ dy \end{pmatrix} = T^T \begin{pmatrix} -ds \\ 0 \\ 0 \end{pmatrix} \tag{327}$$

Converting (327) to three scalar expressions using (304)

$$dx = \cos\tau \cos\nu \, ds \tag{328}$$

$$dy = \cos\tau \sin\nu \, ds \tag{329}$$

$$dz = \sin\tau \, ds \tag{330}$$

The required expression for x is found by integrating
Using η as a dummy variable of integration on s:

$$x(s,t) = -\int_0^s \cos\tau \cos\nu \, d\eta \tag{331}$$

Eq.(330) yields the expression for τ:

$$\sin\tau = dz/ds \triangleq z' \tag{332}$$

From the elementary trigonometric identity $\sin^2 + \cos^2 = 1$,
the cos τ function is seen to be

$$\cos\tau = \sqrt{1 - z'^2} \tag{333}$$

Combining (329) and (332),

$$\sin\nu = -y'/\sqrt{1 - z'^2} \tag{334}$$

Again, from elementary trigonometry,

$$\cos\nu = \sqrt{1 - \frac{y'^2}{1 - z'^2}} \tag{335}$$

Eqs.(332) -(335) are exact expressions which can be used to find T (Eq.(304) in terms of the modal solutions y(s,t) and z(s,t). The angle ζ in (304) is approximately the blade feathering angle (classically denoted $-\theta$), at the point s. θ is a function of blade twist (which makes θ a function of s) and feathering hinge angle. If torsional deformation of the blade exists, θ will also contain the state variables of the torsional dynamic modes.

Eqs.(333) and(335) can be used to eliminate cosτ and cosν from (335):

$$x(s,t) = -\int_0^s \sqrt{1 - z'^2 - y'^2} \, d\eta \tag{336}$$

Eq. (336) allows computation of the blade reference line coordinate x(s,t) directly in terms of the modal solutions y(s,t) and z(s,t). Eq.(336) is possible, because the BRL has been assumed instretchable (see Section B).

Time derivatives of x are required by both the inertial and the aerodynamic distributed loading expressions. These derivative functions can be evaluated directly from (336) by taking the time differentiation inside the integral to operate on the radical. Eq.(336) can be simplified considerably, since z' and y' are relatively small angles. The time differentiation of the approximated version of (336) is also much cleaner than if (336) is used in its exact form. The process of simplifying (336) and than taking the time derivatives of $x(s,t)$ is explained in Appendix I.

III.11 SHAFT LOADS

The load components that are applied to the rotor shaft by a blade can be expressed in integral form. These loads are caused by the presence of the distributed forcing summations (317), (318), (319). Expressed in rotor axis system coordinates, these force and moment components are

$$X_r = \int_0^R p_x \, ds \tag{337}$$

$$Y_r = \int_0^R p_y \, ds \tag{338}$$

$$Z_r = \int_0^R p_z \, ds \tag{339}$$

$$L_r = \int_0^R \left[y p_z - z p_y \right] ds \tag{340}$$

$$M_r = \int_0^R \left[z p_x - x p_z \right] ds \tag{341}$$

$$N_r = \int_0^R \left[x p_y - y p_x \right] ds \tag{342}$$

The integrals in Eqs. (337)-(342) must be evaluated numerically. The blade motion problem must be solved first to determine BRL coordinates x,y,z used in Eqs. (337)-(342). Of course, these coordinates must also be available before the distributed forcing functions can be expressed in numerical form.

IV. MOSTAB COMPUTATIONAL DETAILS

IV.1 INTRODUCTION

This part presents a complete listing for the MOSTAB-B program, written in FORTRAN IV code. The program has been run extensively on a Univac 1108 computer. Typical runs require 1.2 minutes of central processor time, much of which is used for the trim search iteration and input/output operations. No difficulty has been encountered with trim, even for such a complicated helicopter as the AH-56A.

MOSTAB-B requires approximately 42,000 floating point words of digital core. To minimize program development costs, core limits were not considered during the programming phase. As a result, many arrays presently incorporated in the code contain large blocks of zeros. The wasted core space used to store such zero blocks can be eliminated by reprogramming some basic matrix subroutines to handle special matrix configurations. This additional programming effort, plus other core saving measures, can be used to reduce the core required by MOSTAB-B to an estimated size of 25,000-30,000 floating point words.

In addition to the basic MOSTAB-B program described, the code shown below includes several special "convenience" features:

(a) The "repeat run" capability eliminates the need for submitting a complete aircraft description deck for each run. MOSTAB-B executions can be performed sequentially, such that the first input data deck is a complete deck, while subsequent data decks include only the changes in input data required for each case.

(b) Output options allow the MOSTAB-B user to elect to print either all aircraft data stored in the computer, or just the changes and the output for each case.

(c) Input indices specify the peripheral unit numbers to be used for program input and output.

The MOSTAB program development is continuing as described in the main text; therefore, it is suggested that intending users should contact the authors to receive up-to-date information on the current status of the program.

IV.2. LISTING OF MOSTAB 'B' PROGRAM

```
      GROUP 1 (SPEC)             *      GROUP 2 (REDATA)
ITEM            ITEM             *ITEM            ITEM
 1.   WT        11.   LC         * 1.    PK        11.   WE
 2.   XCG       12.   NOPTRM     * 2.    INTS      12.   VNOT
 3.   YCG       13.   IT         * 3.    XEL       13.   PT
 4.   ZCG       14.   QINRTA     * 4.    YEL       14.   PV
 5.   TAS                        * 5.    ZEL       15.   PVDOT
 6.   RHO                        * 6.    A         16.   PF
 7.   PSIDOT                     * 7.    IX        17.   PTC
 8.   PTCHRT                     * 8.    JX        18.   TACPT
 9.   ROLLRT                     * 9.    X         19.   WACPT
10.   HDOT                       * 10.   TE        20.   HACPT

                                                   21.   BDACPT
                                 *
   THE ARRAYS SP AND RED ARE OCCUPYING THE /SPEC/ AND /REDATA/
   COMMON REGIONS.  IN MOSTAB, THESE COMMON REGIONS ARE FILLED
   AS SHOWN BELOW.

   COMMON/SPEC/WT,XCG,YCG,ZCG,TAS,RHO,PSIDOT,PTCHRT,ROLLRT,HDOT,LC,
  1 NOPTRM,IT(6),QINRTA(3,3)
   COMMON/REDATA/PK(250,8),INTS(10,8),XEL(8),YEL(8),ZEL(8),A(6,6,8),
  1 IX(500),JX(500),X(500),TE(6),WE(48),VNOT(9),PT,PV,PVDOT,PF,PTC,
  2 TACPT,WACPT,HACPT,BDACPT

   DIMENSION NTEL(25,2),NDIMS(25,2),RI(7),NSIZE(3,25,2)

   COMMON/SPEC/SP(27)/REDATA/RED(3964)
   COMMON/PHYSCS/PTRANS(250),INTG(10),NPK(8),NINTS(8)
   COMMON/DEBUG/NDXS
   COMMON/IO/IRED,IRYT,IRYTZ

   DATA (NTEL(I,2),I=1,21)/1,2001,2081,2089,2097,2105,2393,2893,
  1 3393,3893,3899,3947,3956,3957,3958,3959,3960,3961,3962,3963,3964/
   DATA (NDIMS(I,1),I=1,14)/12*0,1,2/
   DATA (NDIMS(I,2),I=1,21)/2,2,3*1,3,6*1,9*0/
   DATA ((NSIZE(I,J,1),I=1,3),J=1,14)/36*0,6,0,0,3,3,0/
   DATA ((NSIZE(I,J,2),I=1,3),J=1,21)/250,8,0,10,8,0,8,0,0,8,0,0,8,0,
  1 0,6,6,8,500,0,0,500,0,0,500,0,0,6,0,0,48,0,0,9,0,0,27*0/

   DO 210 I=1,13
   NTEL(I,1)=I
   NTEL(14,1)=19
```

```
READ(5,100)IRED,IRYT,IRYTZ
FORMAT(7I10)
READ(IRED,100) NCASES
WRITE(IRYTZ,11) NCASES
DO 1000 ICASE=1,NCASES

WRITE(IRYTZ,10)
READ(IRED,100) NCHNGS,MSTOPT
WRITE(IRYTZ,11) NCHNGS,MSTOPT
IF(NCHNGS.EQ.0)GO TO 230
DO 225 ICNG=1,NCHNGS

READ(IRED,100) NGROUP,NITEM,NCHNUM,I,J,K
WRITE(IRYTZ,11)NGROUP,NITEM,NCHNUM,I,J,K
FORMAT(1X,7I10)
FORMAT(1H1)

CALCULATE THE START POINT.

NDIM=NDIMS(NITEM,NGROUP)
NST=NTEL(NITEM,NGROUP)
IS=NSIZE(1,NITEM,NGROUP)
JS=NSIZE(2,NITEM,NGROUP)
KS=NSIZE(3,NITEM,NGROUP)
IF(NDIM.EQ.1) NST=NST+I-1
IF(NDIM.EQ.2) NST=NST+I+(J-1)*IS-1
IF(NDIM.EQ.3) NST=NST+I+(J-1)*IS+(K-1)*IS*JS-1

MAKE THE CHANGES.
NLINE=0
NED=NST+NCHNUM-1
DO 300 I=NST,NED

INDEX=I-NST+1
IF(INDEX.LE.NLINE*7) GO TO 270
NLINE=NLINE+1
READ(IRED,120)(RI(J),J=1,7)
WRITE(IRYTZ,12)(RI(J),J=1,7)
FORMAT(7F10.0)
FORMAT(1X,7F14.4)
CONTINUE
NCOL=INDEX-(NLINE-1)*7
IF(NGROUP.EQ.1) SP(I)=RI(NCOL)
IF(NGROUP.EQ.2) RED(I)=RI(NCOL)
CONTINUE
CONTINUE
CONTINUE
CALL MOSTAB(MSTOPT)

CONTINUE
STOP
END
```

```
SUBROUTINE MOSTAB(ISUPRS)

                         MOSTAB

         (MODULAR STABILITY DERIVATIVE PROGRAM)

   INDEX                         DESCRIPTION

   N              NUMBER OF VEHICLE ELEMENTS. (8 OR FEWER NOW)

   NTYPE(I)=      TYPE OF I'TH ELEMENT, WHERE
                1=LIFTING SURFACE
                2=BODY

                3=FLEXIBLE ROTOR (FLEXIBLE BLADES)
                4=RIGID ROTOR

   NTHRU(I)       NUMBER OF TIMES I'TH ELEMENT COMPUTATIONS
                  HAVE BEEN MADE.

   NC(I,1)        NUMBER OF FIRST ELEMENT IN THE CONTROL
                  COLUMN,C, ASSOCIATED WITH VEHICLE ELEMENT I.

   NC(I,2)        TOTAL NUMBER OF ELEMENTS IN C ASSOCIATED WITH
                  AIRCRAFT ELEMENT I.

   NPK(I)         NUMBER OF FLOATING POINT PARAMETERS TO BE READ
                  FOR AIRCRAFT ELEMENT I.

   NIATS(I)       NUMBER OF FIXED POINT PARAMETERS TO BE READ
                  FOR AIRCRAFT ELEMENT I.

   NPASS          NUMBER OF THE PRESENT ITERATION CYCLE.

   LSTPSS       0=GRADIENT MATRICES ARE BEING CALCULATED
                  FOR AN ITERATION CYCLE.
                1=GRADIENT MATRICES ARE BEING CALCULATED FOR
                  STABILITY DERIVATIVE MATRIX CALCULATION.

 INTEGER Z

 DIMENSION [illegible](14)
 DIMENSION B(40,[illegible]),[illegible](40),C(12),S(6),D(6),VA(40),VI(40),VIDOT(40)
 DIMENSION B(40),[illegible](6,6),FT(40,6),FU(40,40),[illegible](40,40),[illegible](40,40)
 DIMENSION UT(40,6),[illegible](40,40),ST(6,6),UC(40,12),CT(12,6)
 DIMENSION [illegible](40,40),[illegible]SAVE(6),[illegible]SAVE(6),FVIDOT(40,40),VVIDOT(40,40)
```

```
      COMMON/IO/IREP,INVT,I
      COMMON/PHYSCS/PTNAME(250),INTS(10),NPX(8),NINTS(8)
      COMMON/ACPRIM/HV1(6),HV2(6),HV3(6)
      COMMON/SPEC/WT,XCG,YCG,ZCG,TAS,RHO,PSIDOT,PTCHRT,ROLLRT,WDOT,LC,
     1 NOPTHV,IT(6),QIMATA(3,3)
      COMMON/REDATA/PX(250,8),INTS(10,8),XEL(8),YEL(8),ZEL(9),A(6,6,8),
     1 IX(500),JX(500),X(500),TE(6),WF(48),VNOT(9),PT,PV,PVDOT,PF,PTC,
     2 TACPT,VACPT,RACPT,RDACPT
      COMMON/COLUMS/CB(12),SB(6),RB(6),VIB(48),VAB(48),FB(48),WB(48),
     1 P(6),DT(6),DW(48)
      COMMON/GRAPHX/FVA(48,48),FVI(48,48),FC(48,12)
      COMMON/INDICS/LC(8,2),LTYPE(8),LTHRU(8),NOPASS, NDIRCT,NEX,NITER
      COMMON/STANFM/FS(6,6),FSDOT(6,6),FC(6,12)
      COMMON/DEBUG/NDBG
      COMMON/HLDMAT/IDHIST,BETA(90,8),BETADT(90,6)

      DATA DCPS/'CB','TE','SB','RB','P','DT','DW','WE','WB','VIB','VAB',
     1 'FB','R','RB'/

      CALL READVT(1,IBUFRS)
      WRITE(6,10)
      FORMAT(1X,8I10)
      FORMAT(1X,8E15.4)

      INITIALIZE.

      [illegible]

      [illegible]
      DO 219 I=1,N
      [illegible]
      DO 217 ISAVE=1,N
      IF(LTYPE(ISAVE).NE.3) GO TO 217
      [illegible](ISAVE)=PX(17,ISAVE)
      [illegible](ISAVE)=PX(18,ISAVE)
      CONTINUE
      NPASS=0
      [illegible]
      DO 300 I=1,NVEL
      [illegible]
      DO 290 J=1,NVEL
      FVI(I,J)=0.0
      FVA(I,J)=0.0
      CONTINUE
```

```
GENERATE THE GEOMETRIC MATRIX G.

DO 250 I=1,N
INAD=6*(I-1)
DO 240 J=1,6
DO 235 K=1,6
NAC=INAD+J
G(NAC,K)=0.0
IF(J.EQ.K) G(NAC,K)=1.0
CONTINUE
CONTINUE
G(INAD+1,5)=ZEL(I)
G(INAD+1,6)=-YEL(I)
G(INAD+2,4)=-ZEL(I)
G(INAD+2,6)=XEL(I)
G(INAD+3,4)=YEL(I)
G(INAD+3,5)=-XEL(I)

CALCULATE SUB-ZERO QUANTITIES.

CONTINUE
NPASS=NPASS+1
IF(NPASS.EQ.1) GO TO 277
DO 474 II=1,6
TH(II)=TS(II)+DT(II)
DO 475 JJ=1,NEL
VV(JJ)=VD(JJ)+DV(JJ)
CONTINUE
CALL CC TH(TS,VVOT,CS)
CALL T LCTV(TS,VVOT,SD,NDIRCT,SPHI,CPHI,STH,CTH)
CALL SC (SD,SPHI,CPHI,STH,CTH,RS)
CALL TAMPV(G,SD,VI0,NEL,6,1,40,6,40)
CALL TADD(VI0,RS,VAR,NEL,1,40,40,40,2)
CALL CHECK(I,1,VAR,VI0,VIRDT,CS,FR,PK,INTS)
CALL A (VAR,VI0,VIRDT,FR,WB,X,IX,JX,A,NEX)

DO 426 I=1,6
P(I)=0.0
DO 422 L=1,NEL
P(I)=P(I)-G(L,I)*FR(L)
CONTINUE

WRITE T  -SEARCH RESULTS IF THIS IS THE FINAL SUB-ZERO
```

```
      COMPUTATION.

      IF(LSTPSS.NE.1) GO TO 283

      WRITE(IPVT,10)
   10 FORMAT(1H1)
      CALL REDRYT(2,ISUPR9)
  283 CONTINUE

      IF(LSTPSS.EQ.1) GO TO 97

      WRITE TRIM-SEARCH RESULTS IN ABBREVIATED FORMAT FOR GENERAL INFO.

      WRITE(2,90) NPASS
   90 FORMAT(10X,'GENERAL TRIM-SEARCH DATA-    PASS NO.',I2)
   91 FORMAT(1X,A3,/,(1X,14F9.3))
   92 FORMAT(/,1X,'FLEXIBLE-BLADE ROTOR,  ELEMENT NO.',I2)
      WRITE(2,91) DOPE(1),(CD(I),I=1,LC)
      WRITE(2,91) DOPE(2),(TF(I),I=1,6)
      WRITE(2,91) DOPE(3),(SD(I),I=1,6)
      WRITE(2,91) DOPE(4),(RD(I),I=1,6)
      WRITE(2,91) DOPE(5),(P(I),I=1,6)
      WRITE(2,91) DOPE(6),(DT(I),I=1,6)
      WRITE(2,91) DOPE(7),(DW(I),I=1,NKEL)
      WRITE(2,91) DOPE(8),(WB(I),I=1,NKEL)
      WRITE(2,91) DOPE(9),(WP(I),I=1,NKEL)
      WRITE(2,91) DOPE(10),(VID(I),I=1,NKEL)
      WRITE(2,91) DOPE(11),(VAD(I),I=1,NKEL)
      WRITE(2,91) DOPE(12),(FD(I),I=1,NKEL)

      DO 95 JD=1,N
      IF(KTYPE(JD).NE.3) GO TO 95
      WRITE(2,92) JD
      NGP=INTS(2,JD)
      WRITE(2,91) DOPE(13),(BETA(I,JD),I=1,NGP)
      WRITE(2,91) DOPE(14),(BETADT(I,JD),I=1,NGP)
   95 CONTINUE
   97 CONTINUE

      INITIALIZE GRADIENT MATRICES

      NDXS=1
      DO 440 I=1,NKEL
      DO 430 J=1,NKEL
      FVI(I,J)=0.0
      WVI(I,J)=0.0
```

```
WRITE TRIM-SEARCH RESULTS IN ABBREVIATED FORMAT FOR GENERAL INFO.

WRITE(2,90) NPASS
FORMAT(10X,'GENERAL TRIM-SEARCH DATA-    PASS NO.',I2)
FORMAT(1X,A3,/,(1X,14F9.3))
FORMAT(/,1X,'FLEXIBLE-BLADE ROTOR,  ELEMENT NO.',I2)
WRITE(2,91) DOPE(1),(CO(I),I=1,LC)
WRITE(2,91) DOPE(2),(TE(I),I=1,6)
WRITE(2,91) DOPE(3),(SD(I),I=1,6)
WRITE(2,91) DOPE(4),(RD(I),I=1,6)
WRITE(2,91) DOPE(5),(P(I),I=1,6)
WRITE(2,91) DOPE(6),(DT(I),I=1,6)
WRITE(2,91) DOPE(7),(DW(I),I=1,NKEL)
WRITE(2,91) DOPE(8),(WF(I),I=1,NKEL)
WRITE(2,91) DOPE(9),(WD(I),I=1,NKEL)
WRITE(2,91) DOPE(10),(VID(I),I=1,NKEL)
WRITE(2,91) DOPE(11),(VAD(I),I=1,NKEL)
WRITE(2,91) DOPE(12),(FD(I),I=1,NKEL)

DO 95 JD=1,N
IF(NTYPE(JD).NE.3) GO TO 95
WRITE(2,92) JD
NRP=INTS(2,JD)
WRITE(2,91) DOPE(13),(BETA(I,JD),I=1,NRP)
WRITE(2,91) DOPE(14),(BETADT(I,JD),I=1,NRP)
CONTINUE
CONTINUE

INITIALIZE GRADIENT MATRICES

NOXS=1
DO 440 I=1,NKEL
DO 439 J=1,NKEL
FVI(I,J)=0.0
WVI(I,J)=0.0
FVA(I,J)=0.0
WVA(I,J)=0.0
FVIDT(I,J)=0.0
WVIDT(I,J)=0.0
WF(I,J)=0.0
IF(J.GT.LC) GO TO 439
FC(I,J)=0.0
WC(I,J)=0.0
CONTINUE
CONTINUE

CALCULATE THE GRADIENT MATRICES. FIRST, FIND CT, ST AND RT.

IF(NSTEPS.EQ.1) GO TO 345
```

```
DO 310 I=1,6
TE(I)=TE(I)+PT
CALL CONTRL(TE,VNOT,C)
CALL VELCTY(TE,VNOT,S,NDIRCT,SPHI,CPHI,STH,CTH)
CALL FCERGD(S,SPHI,CPHI,STH,CTH,R)
DO 308 J=1,LC
CT(J,I)=(C(J)-CO(J))/PT
DO 307 J=1,6
ST(J,I)=(S(J)-SO(J))/PT
RT(J,I)=(R(J)-RO(J))/PT
TE(I)=TE(I)-PT
CONTINUE

FIND FVI AND WVI

DO 320 K=1,N
DO 319 I=1,6
IRO=6*(K-1)+I
VIO(IRO)=VIO(IRO)+PV
CALL FORCE(K,2,VAO,VIO,VIDOT,CO,F,PK,INTS)
CALL WASH(VAO,VIO,VIDOT,FO,W,X,IX,JX,A,NEX)
DO 318 L=1,6
LRO=6*(K-1)+L
FVI(LRO,IRO)=(F(LRO)-FO(LRO))/PV
DO 317 L=1,NKEL
WVI(L,IRO)=(W(L)-WO(L))/PV
VIO(IRO)=VIO(IRO)-PV
CONTINUE

CALCULATE FVA AND WVA

DO 330 K=1,N
DO 329 I=1,6
IRO=6*(K-1)+I
VAO(IRO)=VAO(IRO)+PV
CALL FORCE(K,2,VAO,VIO,VIDOT,CO,F,PK,INTS)
CALL WASH(VAO,VIO,VIDOT,FO,W,X,IX,JX,A,NEX)
DO 328 L=1,6
LRO=6*(K-1)+L
FVA(LRO,IRO)=(F(LRO)-FO(LRO))/PV
DO 327 L=1,NKEL
WVA(L,IRO)=(W(L)-WO(L))/PV
VAO(IRO)=VAO(IRO)-PV
CONTINUE

COMPUTE FC AND WC.  COMPUTE ELEMENTS OF F ONLY FOR THOSE VEHICLE
```

```
TE(I)=TF(I)-PT
CONTINUE

FIND FVI AND WVI

DO 320 K=1,N
DO 319 I=1,6
IRO=6*(K-1)+I
VIO(IRO)=VIO(IRO)+PV
CALL FORCE(K,2,VAO,VIO,VIDOT,CO,F,PK,INTS)
CALL WASH(VAO,VIO,VIDOT,FO,W,X,IX,JX,A,NEX)
DO 318 L=1,6
LRO=6*(K-1)+L
FVI(LRO,IRO)=(F(LRO)-FO(LRO))/PV
DO 317 L=1,NKEL
WVI(L,IRO)=(W(L)-WO(L))/PV
VIO(IRO)=VIO(IRO)-PV
CONTINUE

CALCULATE FVA AND WVA

DO 330 K=1,N
DO 329 I=1,6
IRO=6*(K-1)+I
VAO(IRO)=VAO(IRO)+PV
CALL FORCE(K,2,VAO,VIO,VIDOT,CO,F,PK,INTS)
CALL WASH(VAO,VIO,VIDOT,FO,W,X,IX,JX,A,NEX)
DO 328 L=1,6
LRO=6*(K-1)+L
FVA(LRO,IRO)=(F(LRO)-FO(LRO))/PV
DO 327 L=1,NKEL
WVA(L,IRO)=(W(L)-WO(L))/PV
VAO(IRO)=VAO(IRO)-PV
CONTINUE

COMPUTE FC AND WC.  COMPUTE ELEMENTS OF F ONLY FOR THOSE VEHICLE
COMPONENTS AFFECTED BY C(J).

DO 340 J=1,LC
CO(J)=CO(J)+PTC
DO 338 K=1,N
IF(NC(K,2))332,338,332
IF(J.LT.NC(K,1).OR.J.GE.(NC(K,1)+NC(K,2))) GO TO 338
CALL FORCE(K,2,VAO,VIO,VIDOT,CO,F,PK,INTS)
DO 334 L=1,6
LRO=6*(K-1)+L
FC(LRO,J)=(F(LRO)-FO(LRO))/PTC
CONTINUE
CALL WASH(VAO,VIO,VIDOT,FO,W,X,IX,JX,A,NEX)
DO 339 L=1,NKEL
```

```
WC(L,J)=(W(L)-W0(L))/DTC
C0(J)=C0(J)-DTC

FIND WF

DO 348 I=1,NKFL
F0(I)=F0(I)+DF
CALL WASH(VA0,VI0,VIDOT,F0,W,X,IX,JX,A,NEX)
DO 347 L=1,NKFL
WF(L,I)=(W(L)-W0(L))/DF
F0(I)=F0(I)-DF

COMPUTE FVIDOT AND WVIDOT.

IF(LSTPSS.NE.1)GO TO 361
DO 360 K=1,N
DO 359 I=1,6
IRO=6*(K-1)+I
VIDOT(IRO)=PVDOT
CALL FORCE(K,2,VA0,VI0,VIDOT,C0,F,PK,INTS)
CALL WASH(VA0,VI0,VIDOT,F0,W,X,IX,JX,A,NEX)
DO 358 L=1,6
LRO=6*(K-1)+L
FVIDOT(LRO,IRO)=(F(LRO)-F0(LRO))/PVDOT
DO 357 L=1,NKFL
WVIDOT(L,IRO)=(W(L)-W0(L))/PVDOT
VIDOT(IRO)=VIDOT(IRO)-PVDOT
CONTINUE
CONTINUE

THE GRADIENT MATRICES ARE AVAILABLE.  NOW SOLVE FOR DT AND DW.

IF(LSTPSS.EQ.1) GO TO 450
K=NKFL
CALL MTXMPY(WC,CT,VT,K,LC,6,48,12,48)
CALL MTXMPY(G,ST,WC,K,6,6,48,6,48)
CALL MTXMPY(WVI,WC,VW,K,K,6 ,48,48,48)
CALL MTXADD(VW,VT,WVI,K,6,48,48,48,1)
CALL MTXMPY(WF,FVA,FW,K,K,K,48,48,48)
CALL MTXADD(FW,WVA,VW,K,K,48,48,48,1)
DO 380 I=1,K
VW(I,I)=VW(I,I)+1.0
CALL MATINV(VW,48,K,DET,IRANK)

IF(IRANK.EQ.K) GO TO 381
WRITE(2,830) DET,IRANK
```

```
FO(I)=FC(I)-PF

COMPUTE FVIDOT AND WVIDOT.

IF(LSTPSS.NE.1)GO TO 361
DO 360 K=1,N
DO 359 I=1,6
IRO=6*(K-1)+I
VIDOT(IRO)=PVDOT
CALL FORCE(K,2,VAO,VIO,VIDOT,CO,F,PK,INTS)
CALL WASH(VAO,VIO,VIDOT,FO,W,X,IX,JX,A,NEX)
DO 358 L=1,6
LRO=6*(K-1)+L
FVIDOT(LRO,IRO)=(F(LRO)-FO(LRO))/PVDOT
DO 357 L=1,NKEL
WVIDOT(L,IRO)=(W(L)-WO(L))/PVDOT
VIDOT(IRO)=VIDOT(IRO)-PVDOT
CONTINUE
CONTINUE

THE GRADIENT MATRICES ARE AVAILABLE.  NOW SOLVE FOR DT AND DW.

IF(LSTPSS.EQ.1) GO TO 450
K=NKEL
CALL MTXMPY(WC,CT,VT,K,LC,6,48,12,48)
CALL MTXMPY(G,ST,WC,K,6,6,48,6,48)
CALL MTXMPY(WVI,WC,VW,K,K,6 ,48,48,48)
CALL MTXADD(VW,VT,WVI,K,6,48,48,48,1)
CALL MTXMPY(WF,FVA,FW,K,K,K,48,48,48)
CALL MTXADD(FW,WVA,VW,K,K,48,48,48,1)
DO 380 I=1,K
VW(I,I)=VW(I,I)+1.0
CALL MATINV(VW,48,K,DET,IRANK)

IF(IRANK.EQ.K) GO TO 381
WRITE(7,830) DET,IRANK
GO TO 1000
CONTINUE

CALL MTXMPY(FVI,WC,FW,K,K,6,48,48,48)
CALL MTXMPY(FC,CT,FVI,K,LC,6,48,12,48)
CALL MTXADD(FVI,FW,FC,K,6,48,48,48,1)
CALL MTXMPY(WF,FC,FVI,K,K,6,48,48,48)
CALL MTXADD(WC,FVI,VT,K,6,48,48,48,2)
CALL MTXADD(VT,WVI,FVI,K,6,48,48,48,2)
CALL MTXMPY(VW,FVI,VT,K,K,6,48,48,48)
CALL MTXMPY(FVA,VT,FVI,K,K,6,48,48,48)
CALL MTXADD(FVI,FC,FT,K,6,48,48,48,1)
CALL MTXMPY(FVA,VW,FW,K,K,K,48,48,48)
DO 385 ICP=1,6
```

```
DO 384 KCP=1,6
TEMP=0.0
DO 383 LCP=1,K
TEMP=TEMP+G(LCP,JCP)*FT(LCP,KCP)
RT(JCP,KCP)=RT(JCP,KCP)-TEMP
CONTINUE
CALL MATINV(RT,6,6,DET,IRANK)

IF(IRANK.EQ.6) GO TO 387
WRITE(7,630) DET,IRANK
GO TO 1000
FORMAT(//,1X,'INVERSION FLAG',5X,'DETERMINANT=',E11.4,5X,
. 'RANK=',I2,//)
CONTINUE

CALL MTXADD(WO,WE,W,K,1,48,48,48,2)
CALL MTXMPY(FW,W,VI,K,K,1,48,48,48)
CALL MTXADD(FO,VI,F,K,1,48,48,48,2)
DO 395 JCP=1,6
RV1(JCP)=0.0
DO 393 LCP=1,K
RV1(JCP)=RV1(JCP)+G(LCP,JCP)*F(LCP)
RV2(JCP)=RV1(JCP)-R0(JCP)
CALL MTXMPY(RT,RV2,DT,6,6,1,6,6,6)

DT IS AVAILABLE.  NOW FIND DW.

CALL MTXMPY(VW,W,F,K,K,1,48,48,48)
CALL MTXMPY(WVA,F,VI,K,K,1,48,48,48)
CALL MTXMPY(FW,W,F,K,K,1,48,48,48)
CALL MTXMPY(WF,F,VA,K,K,1,48,48,48)
CALL MTXADD(VI,VA,F,K,1,48,48,48,1)
CALL MTXMPY(WVI,DT,VI,K,6,1,48,6,48)
CALL MTXMPY(VT,DT,VA,K,6,1,48,6,48)
CALL MTXMPY(WVA,VA,W,K,K,1,48,48,48)
CALL MTXADD(W,VI,VA,K,1,48,48,48,1)
CALL MTXMPY(FT,DT,VI,K,6,1,48,6,48)
CALL MTXMPY(WF,VI,W,K,K,1,48,48,48)
CALL MTXADD(W,VA,VI,K,1,48,48,48,1)
CALL MTXADD(VI,F,DW,K,1,48,48,48,2)

FIND MODULI  OF CONVERGENCE.

WCLOS=0.0
TCLOS=0.0
BCLOS=0.0
```

```
      GO TO 1000
      FORMAT(//,1X,'INVERSION FLAG',5X,'DETERMINANT=',E11.4,5X,
     1 'RANK=',I2,//)
      CONTINUE

      CALL MTXADD(WD,WE,W,K,1,48,48,48,2)
      CALL MTXMPY(FW,W,VI,K,K,1,48,48,48)
      CALL MTXADD(FD,VI,F,K,1,48,48,48,2)
      DO 395 JCP=1,6
      RV1(JCP)=0.0
      DO 393 LCP=1,K
      RV1(JCP)=RV1(JCP)+G(LCP,JCP)*F(LCP)
      RV2(JCP)=RV1(JCP)-RD(JCP)
      CALL MTXMPY(RT,RV2,DT,6,6,1,6,6,6)

C     DT IS AVAILABLE.  NOW FIND DW.

      CALL MTXMPY(VW,W,F,K,K,1,48,48,48)
      CALL MTXMPY(WVA,F,VI,K,K,1,48,48,48)
      CALL MTXMPY(FW,W,F,K,K,1,48,48,48)
      CALL MTXMPY(WF,F,VA,K,K,1,48,48,48)
      CALL MTXADD(VI,VA,F,K,1,48,48,48,1)
      CALL MTXMPY(WVI,DT,VI,K,6,1,48,6,48)
      CALL MTXMPY(VT,DT,VA,K,6,1,48,6,48)
      CALL MTXMPY(WVA,VA,W,K,K,1,48,48,48)
      CALL MTXADD(W,VI,VA,K,1,48,48,48,1)
      CALL MTXMPY(FT,DT,VI,K,6,1,48,6,48)
      CALL MTXMPY(WF,VI,W,K,K,1,48,48,48)
      CALL MTXADD(W,VA,VI,K,1,48,48,48,1)
      CALL MTXADD(VI,F,DW,K,1,48,48,48,2)

C     FIND MODULI  OF CONVERGENCE.

      WCLOS=0.0
      TCLOS=0.0
      BCLOS=0.0
      BDCLOS=0.0
      DO 405 ICL=1,NKEL
      TEMP=ABS(DW(ICL))
      WCLOS=WCLOS+TEMP
      DO 406 ICL=1,6
      TEMP=ABS(DT(ICL))
      TCLOS=TCLOS+TEMP
      DO 408 ICL=1,N
      IF(MTYPE(ICL).NE.3)GO TO 408
      TEMPH=ABS(PK(36,ICL)-BSAVE(ICL))
      TEMPHD=ABS(PK(37,ICL)-BDSAVE(ICL))
      BCLOS=BCLOS+TEMPH
      BDCLOS=BDCLOS+TEMPHD
      BSAVE(ICL)=PK(36,ICL)
```

```
BDSAVE(ICL)=PK(37,ICL)
CONTINUE

DETERMINE IF THE PROXIMITY TO A TRIM SOLUTION IS ACCEPTABLE, OR IF
ANOTHER ITERATION CYCLE IS ALLOWABLE, AND TAKE THE PROPER ACTION.

IF(WCLCS-WACPT)412,412,420
IF(TCLCS-TACPT)413,413,420
IF(RCLCS-RACPT)414,414,420
IF(RDCLCS-RDACPT)435,435,420
IF(NPASS-NITER)275,427,427
WRITE(IRYT,10)
WRITE(IRYT,428)
FORMAT(//,1X,'AN ACCEPTABLE TRIM SOLUTION HAS NOT BEEN FOUND.')
DO 808 I=1,6
P(I)=0.0
DO 806 L=1,NKEL
P(I)=P(I)+G(L,I)*FO(L)
CONTINUE
CALL REDRYT(2,ISUPRS)
GO TO 1000
CONTINUE

A TRIM SOLUTION IS AVAILABLE AS TE AND WE.PROCEED TO FIND THE
STABILITY DERIVATIVE MATRICES.

LSTPSS=1
NDXS=1
GO TO 275
CONTINUE

WRITE INTERESTING ITEMS.

CALL REDRYT(3,ISUPRS)

MANIPULATE GRADIENT MATRICES TO GET STABILITY DERIVATIVE MATRICES.

K=NKEL
CALL MTXMPY(WF,FVI,VW,K,K,K,48,48,48)
CALL MTXADD(WVI,VW,FW,K,K,48,48,48,1)
DO 710 I=1,K
FW(I,I)=FV(I,I)-1.0
CALL MTXMPY(WF,FVA,WVI,K,K,K,48,48,48)
CALL MTXADD(WVA,WVI,VW,K,K,48,48,48,1)
DO 715 I=1,K
VW(I,I)=VW(I,I)+1.0
```

```
IF(NPASS-NITER)275,427,427
WRITE(IRYT,10)
WRITE(IRYT,428)
FORMAT(//,1X,'AN ACCEPTABLE TRIM SOLUTION HAS NOT BEEN FOUND.')
DO 408 I=1,6
P(I)=0.0
DO 408 L=1,NKEL
P(I)=P(I)+G(L,I)*FO(L)
CONTINUE
CALL REDRYT(2,ISUPRS)
GO TO 1000
CONTINUE

A TRIM SOLUTION IS AVAILABLE AS TE AND WE.PROCEED TO FIND THE
STABILITY DERIVATIVE MATRICES.

LSTPSS=1
MDKS=1
GO TO 275
CONTINUE

WRITE INTERESTING ITEMS.

CALL REDRYT(3,ISUPRS)

MANIPULATE GRADIENT MATRICES TO GET STABILITY DERIVATIVE MATRICES.

K=NKEL
CALL MTXMPY(WF,FVI,VW,K,K,K,48,48,48)
CALL MTXADD(WVI,VW,FW,K,K,48,48,48,1)
DO 710 I=1,K
FW(I,I)=FW(I,I)-1.0
CALL MTXMPY(WF,FVA,WVI,K,K,K,48,48,48)
CALL MTXADD(WVA,WVI,VW,K,K,48,48,48,1)
DO 715 I=1,K
VW(I,I)=VW(I,I)+1.0
CALL MATINV(VW,48,K,DET,IRANK)
CALL MTXMPY(VW,FW,WVI,K,K,K,48,48,48)
CALL MTXMPY(FVA,WVI,FW,K,K,K,48,48,48)
CALL MTXADD(FVI,FW,WVI,K,K,48,48,48,2)
CALL MTXMPY(WVI,G,FW,K,K,6,48,48,48)
DO 718 I=1,6
DO 717 J=1,6
PS(I,J)=0.0
DO 716 L=1,K
PS(I,J)=PS(I,J)+G(L,I)*FW(L,J)
CONTINUE
CONTINUE

FORMAT(///)
```

```
      CALL MTXMPY(WF,FVIDOT,FV,K,K,K,48,48,48)
      CALL MTXADD(FV,WVIDOT,WVI,K,K,48,48,48,1)
      CALL MTXMPY(VX,WVI,FW,K,K,K,48,48,48)
      CALL MTXMPY(FVA,FW,WVI,K,K,K,48,48,48)
      CALL MTXADD(FVIDOT,WVI,FW,K,K,48,48,48,2)
      CALL MTXMPY(FW,G,WVI,K,K,6,48,48,48)
      DO 728 I=1,6
      DO 727 J=1,6
      PSDOT(I,J)=0.0
      DO 726 L=1,K
      PSDOT(I,J)=PSDOT(I,J)+G(L,I)*WVI(L,J)
      CONTINUE
      CONTINUE

      CALL MTXMPY(WF,FC,FW,K,K,LC,48,48,48)
      CALL MTXADD(FW,WC,WVI,K,LC,48,48,48,1)
      CALL MTXMPY(VX,WVI,FW,K,K,LC,48,48,48)
      CALL MTXMPY(FVA,FW,WVI,K,K,LC,48,48,48)
      CALL MTXADD(FC,WVI,FW,K,LC,48,48,48,2)
      DO 738 I=1,6
      DO 737 J=1,LC
      PC(I,J)=0.0
      DO 736 L=1,K
      PC(I,J)=PC(I,J)+G(L,I)*FW(L,J)
      CONTINUE
      CONTINUE

      WRITE(IWRT,20)
      FORMAT(1H1,1X,'STABILITY DERIVATIVES WITH RESPECT TO OVERALL ',
     1 'VEHICLE CO-ORDINATES.')
      CALL HECRVT(4,ISUPRS)

      CONVERT STABILITY DERIVATIVES TO STABILITY AXIS SYSTEM COORDINATES

      UCG=SD(1)+ZCG*SD(5)-YCG*SD(6)
      VCG=SD(2)+XCG*SD(6)-ZCG*SD(4)
      WCG=SD(3)+YCG*SD(4)-XCG*SD(5)
      SPEED=SQRT(UCG**2+VCG**2+WCG**2)
      RDCL=SQRT(UCG**2+WCG**2)
      IF(SPEED)748,748,750
      IF(RDCL)748,748,751
      CONTINUE
      WRITE(IWRT,749)
      FORMAT(//,1X,'STABILITY AXES ARE UNDEFINED BECAUSE ',
     1 'THE INERTIAL SPEED IS ZERO')
```

```
      DO 700 L=1,6
      PSDOT(I,J)=PSDOT(I,J)+S(L,I)*DVI(L,J)
      CONTINUE
      CONTINUE

      CALL MTXMPY(DF,FC,FU,K,K,LC,48,48,48)
      CALL MTXADD(FU,FC,DVI,K,LC,48,48,48,1)
      CALL MTXMPY(VM,DVI,FU,K,K,LC,48,48,48)
      CALL MTXMPY(FVA,FU,DVI,K,K,LC,48,48,48)
      CALL MTXADD(FP,DVI,FU,K,LC,48,48,48,2)
      DO 738 I=1,6
      DO 737 J=1,6
      PC(I,J)=0.0
      DO 736 L=1,6
      PC(I,J)=PC(I,J)+S(L,I)*FU(L,J)
      CONTINUE
      CONTINUE

      WRITE(IOUT,78)
      FORMAT(1H1,1X,'STABILITY DERIVATIVES WITH RESPECT TO OVERALL ',
     1 'VEHICLE CO-ORDINATES.')
      CALL DERPRT(4,IRUPRS)

      CONVERT STABILITY DERIVATIVES TO STABILITY AXIS SYSTEM COORDINATES

      UCG=SD(1)+ZCG*SD(5)-YCG*SD(6)
      VCG=SD(2)+XCG*SD(6)-ZCG*SD(4)
      WCG=SD(3)+YCG*SD(4)-XCG*SD(5)
      SPEED=SQRT(UCG**2+VCG**2+WCG**2)
      RPCL=SQRT(UCG**2+WCG**2)
      IF(SPEED)748,748,750
      IF(RPCL)748,748,751
      CONTINUE
      WRITE(IOUT,749)
      FORMAT(//,1X,'STABILITY AXES ARE UNDEFINED BECAUSE ',
     1 'THE INERTIAL SPEED IS ZERO')
      GO TO 1000
      CONTINUE
      CTH=UCG/RPCL
      STH=-WCG/RPCL
      CCV=RPCL/SPEED
      SCV=VCG/SPEED

      RT(1,1)=CTH*CCV
      RT(1,2)=SCV
      RT(1,3)=-CCV*STH
      RT(2,1)=-SCV*CTH
      RT(2,2)=CCV
      RT(2,3)=SCV*STH
      RT(3,1)=STH
```

```
      RT(3,2)=0.0
      RT(3,3)=CTW
C
C     ASSEMBLE THE STABILITY DERIVATIVE TRANSFER MATRIX.
C
      DO 755 I=1,3
      DO 754 J=1,3
      ST(I,J)=RT(I,J)
      ST(I,J+3)=0.0
      ST(I+3,J+3)=RT(I,J)
      ST(I+3,1)=YCG*RT(I,3)-ZCG*RT(I,2)
      ST(I+3,2)=ZCG*RT(I,1)-XCG*RT(I,3)
      ST(I+3,3)=XCG*RT(I,2)-YCG*RT(I,1)
      CONTINUE
C
C     ROTATE THE VEHICLE INERTIA TENSOR TO STABILITY AXES.
C
      CALL MTXMPY(RT,QINRTA,WF,3,3,3,6,3,48)
      DO 760 I=1,3
      DO 759 J=1,3
      FT(I,J)=RT(J,I)
      CONTINUE
      CALL MTXMPY(WF,FT,FW,3,3,3,48,48,48)
C
C     WRITE THE ROTATED INERTIA TENSOR
C
      WRITE(IPYT,79)
      WRITE(IPYT,13)((FW(I,J),J=1,3),I=1,3)
      FORMAT(//,1X,'THE INERTIA TENSOR EXPRESSED WITH RESPECT TO ',
     1 'STABILITY AXES-',/,3(/,10X,3E15.4),/)
C
C     ROTATE STABILITY DERIVATIVE MATRICES TO STABILITY AXES.
C
      CALL MTXMPY(ST,PS,FVI,6,6,6,6,6,48)
      CALL MTXMPY(ST,PSDOT,FVIDOT,6,6,6,6,6,48)
      CALL MTXMPY(ST,PC,FC,6,6,LC,6,6,48)
      DO 765 I=1,6
      DO 764 J=1,6
      RT(I,J)=ST(J,I)
      CONTINUE
      CALL MTXMPY(FVI,RT,PS,6,6,6,48,6,6)
      CALL MTXMPY(FVIDOT,RT,PSDOT,6,6,6,48,6,6)
      DO 767 I=1,6
      DO 766 J=1,LC
      PC(I,J)=FC(I,J)
      CONTINUE
```

```
      ST(I+3,1)=YCGORT(I,3)-ZCGORT(I,2)
      ST(I+3,2)=ZCGORT(I,1)-XCGORT(I,3)
      ST(I+3,3)=XCGORT(I,2)-YCGORT(I,1)
      CONTINUE

      ROTATE THE VEHICLE INERTIA TENSOR TO STABILITY AXES.

      CALL MTXMPY(RT,QINRTA,WF,3,3,3,6,3,48)
      DO 760 I=1,3
      DO 759 J=1,3
      FT(I,J)=RT(J,I)
      CONTINUE
      CALL MTXMPY(WF,FT,FW,3,3,3,48,48,48)

      WRITE THE ROTATED INERTIA TENSOR

      WRITE(IRYT,79)
      WRITE(IRYT,13)((FW(I,J),J=1,3),I=1,3)
      FORMAT(//,1X,'THE INERTIA TENSOR EXPRESSED WITH RESPECT TO ',
     1 'STABILITY AXES-',/,3(/,10X,3E15.4),/)

      ROTATE STABILITY DERIVATIVE MATRICES TO STABILITY AXES.

      CALL MTXMPY(ST,PS,FVI,6,6,6,6,6,48)
      CALL MTXMPY(ST,PSDOT,FVIDOT,6,6,6,6,6,48)
      CALL MTXMPY(ST,PC,FC,6,6,LC,6,6,48)
      DO 765 I=1,6
      DO 764 J=1,6
      RT(I,J)=ST(J,I)
      CONTINUE
      CALL MTXMPY(FVI,RT,PS,6,6,6,48,6,6)
      CALL MTXMPY(FVIDOT,RT,PSDOT,6,6,6,48,6,6)
      DO 767 I=1,6
      DO 766 J=1,LC
      PC(I,J)=FC(I,J)
      CONTINUE

      WRITE THE STABILITY AXIS ARRAYS.

      WRITE(IRYT,21)
      FORMAT(1H1,'STABILITY DERIVATIVES WITH RESPECT TO ',
     1 'STABILITY AXES (DIMENSIONAL),')
      CALL REFRYT(4,ISUPRS)

      PUNCH THE STABILITY DERIVATIVES, INERTIAS AND MISC. TRIM ITEMS.

      PUNCH 810
      PUNCH 810,((PS(I,J),J=1,6),I=1,6)
      PUNCH 810,((PSDOT(I,J),J=1,6),I=1,6)
      DO 769 I=1,6
```

```
      PUNCH 810,(PC(I,J),J=1,LC)
      DO 811 I=1,3
      PUNCH 810,(FW(I,J),J=1,3)
      PUNCH 810,(SD(I),I=1,6),(TE(J),J=1,6)
      FORMAT(6F12.4)

      DIVIDE THROUGH THE MATRICES BY THE MASS AND INERTIAS.

      QMASS=WT/32.2
      DO 775 I=1,6
      DO 774 J=1,LC
      IF(I.GT.3) GO TO 772
      IF(J.GT.6) GO TO 771
      PS(I,J)=PS(I,J)/QMASS
      PSDOT(I,J)=PSDOT(I,J)/QMASS
      PC(I,J)=PC(I,J)/QMASS
      GO TO 774
      CONTINUE
      IF(J.GT.6) GO TO 773
      PS(I,J)=PS(I,J)/FW(I-3,I-3)
      PSDOT(I,J)=PSDOT(I,J)/FW(I-3,I-3)
      PC(I,J)=PC(I,J)/FW(I-3,I-3)
      CONTINUE
      CONTINUE

      WRITE THE STABILITY AXIS ARRAYS.

      WRITE(IRYT,22)
      FORMAT(1H1,'STABILITY DERIVATIVES WITH RESPECT TO ',
     1 'STABILITY AXES',/,1X,'(DIVIDED BY THE INERTIAS).')
      CALL REDRYT(4,ISUPRS)
      CONTINUE
      RETURN
      END
```

```
      SUBROUTINE REDRYT(IPHASE,ISU)

      INTEGER R,W
      DIMENSION COLS(6,2),ELTYP(16),ROWS(6),CLABEL(6),GITLE(20)

      COMMON/IO/R,W
      COMMON/STABDR/STD(6,6,2),PC(6,12)
      COMMON/TITLES/TITLE(20,8)
      COMMON/PHYSCS/PTRANS(250),INTG(10),NPK(8),NINTS(8)
      COMMON/COLUMS/CO(12),SO(6),RO(6),VIO(48),VAO(48),FO(48),WO(48),
     1 P(6),DT(6),DW(48)
      COMMON/GRADMX/FVA(48,48),FVI(48,48),FC(48,12)
      COMMON/INDECS/NC(8,2),NTYPE(8),NTHRU(8),N,NPASS, NDIRCT,NEX,NITER
      COMMON/SPEC/WT,XCG,YCG,ZCG,TAS,RHO,PSIDOT,PTCHRT,ROLLRT,HDOT,LC,
     1 NOPTRM,IT(6),QINRTA(3,3)
      COMMON/BLDHST/IOHIST,BETA(50,8),BETADT(50,8)
      COMMON/REDATA/PK(250,8),INTS(10,8),XEL(8),YEL(8),ZEL(8),A(6,6,8),
     1 IX(500),JX(500),X(500),TE(6),WE(48),VNOT(9),PT,PV,PVDOT,PF,PTC,
     2 TACPT,WACPT,RACPT,RDACPT

      DATA ELTYP/'LIFTI','NG SU','RFACE',' ',

     1 'AEROD','YNAMI','C BOD','Y',
     2 'ROTOR',' (FLE','X. BL','ADES)',
     3 'ROTOR',' (RIG','ID BL','ADES)'/
      DATA ROWS/'X','Y','Z','L','M','N'/
      DATA CLABEL/'C(  )','C(  )','C(  )','C(  )','C(  )','C(  )'/
      DATA COLS/' U ',' V ',' W ',' P ',' Q ',' R ',
     1 'U DOT','V DOT','W DOT','P DOT','Q DOT','R DOT'/

      GO TO(501,502,503,504),IPHASE
      CONTINUE

      WRITE(W,10)
      DO 12 ITITL=1,6
      READ(R,100)(GITLE(I),I=1,20)
      WRITE(W,11)(GITLE(I),I=1,20)
      IF(ISU.EQ.1) GO TO 999
      FORMAT(10A4,10A3)
      FORMAT(1H1,10A4,10A3)
      FORMAT(1X,10A4,10A3)

      IF(ISU.EQ.2) GO TO 602
      READ(R,110) N,NOPTRM,NDIRCT,NEX,NITER
      READ(R,110)(NTYPE(I),I=1,N)
      READ(R,110)(NC(I,1),NC(I,2),I=1,N)
      READ(R,110)(IT(I),I=1,6)
      CONTINUE
      FORMAT(8I10)
```

```
LC=0
DO 2 I=1,N
LC=LC+NC(I,2)

WRITE(W,20)N
FORMAT(1H0,'NUMBER OF VEHICLE ELEMENTS (N)=',I2,5X,'*',
1 13X,'INFORMATION',/,39X,'*')

WRITE(W,21)NOPTRM
FORMAT(1X,'TRIM OPTION INDEX (NOPTRM)=',I2,10X,'A. NOPTRM SPEC',
1 'IFIES THE PROGRAM',/,1X,'(SEE INFORMATION-A)',19X,'*   VARIAB',
2 'LES REPRESENTED BY THE',/,39X,'*   INPUT QUANTITIES PTCHRT ',
3 '(PITCH')

WRITE(W,22)NDIRCT
FORMAT(1X,'FLIGHT DIRECTION (NDIRCT)=',I2,10X,'*   RATE) AND ',
1 'ROLLRT (ROLL RATE)-',/,1X,'(NDIRCT=0 FOR FORWARD',17X,'*',/,
2 2X,'FLIGHT OR 1 FOR BACKWARD FLIGHT)',5X,'*',19X,'QUANTITIES',/,

WRITE(W,10)
DO 12 ITITL=1,6
READ(R,100)(GITLE(I),I=1,20)
WRITE(W,11)(GITLE(I),I=1,20)
IF(ISU.EQ.1) GO TO 999
FORMAT(10A4,10A3)
FORMAT(1H1,10A4,10A3)
FORMAT(1X,10A4,10A3)

IF(ISU.EQ.2) GO TO 602
READ(R,110) N,NOPTRM,NDIRCT,NEX,NITER
READ(R,110)(NTYPE(I),I=1,N)
READ(R,110)(NC(I,1),NC(I,2),I=1,N)
READ(R,110)(IT(I),I=1,6)
CONTINUE
FORMAT(8I10)
```


```
LC=0
DO 2 I=1,N
LC=LC+NC(I,2)

WRITE(W,20)N
FORMAT(1H0,'NUMBER OF VEHICLE ELEMENTS (N)=',I2,5X,'*',
1 13X,'INFORMATION',/,39X,'*')

WRITE(W,21)NOPTRM
FORMAT(1X,'TRIM OPTION INDEX (NOPTRM)=',I2,10X,'A. NOPTRM SPEC',
1 'IFIES THE PROGRAM',/,1X,'(SEE INFORMATION-A)',19X,'*   VARIAB',
2 'LES REPRESENTED BY THE',/,39X,'*   INPUT QUANTITIES PTCHRT ',
3 '(PITCH')
```

```
      WRITE(W,22)NDIRCT
      FORMAT(1X,'FLIGHT DIRECTION (NDIRCT)=',I2,10X,'*    RATE) AND ',
     1 'ROLLRT (ROLL RATE)-',/,1X,'(NDIRCT=0 FOR FORWARD',17X,'*',/,
     2 2X,'FLIGHT OR 1 FOR BACKWARD FLIGHT)',5X,'*',19X,'QUANTITIES',/,
     3 39X,'*',6X,'VALUE',8X,'DEFINED BY',/,1X,'NUMBER OF ELEMENTS ',
     4 'TO BE READ INTO',4X,'*',7X,'OF',10X,'PTCHRT AND',/,1X,'THE ',
     5 'INTERFERENCE VELOCITY COUPLING',4X,'*',5X,'NOPTHM',10X,'ROLLRT')

      WRITE(W,23)NEX
      FORMAT(1X,'MATRIX,X, (NEX)=',I2,20X, '*',/,39X,'*',8X,'1', 7X,
     1 'THETA DOT, PHI DOT',/,1X,'MAXIMUM ALLOWABLE NUMBER OF TRIM',
     2 6X,'*',8X,'2',7X,'THETA DOT,P')

      WRITE(W,24)NITER
      FORMAT(1X,'ITERATION CYCLES (NITER)=',I2,11X, '*',8X,'3',7X,
     1 'Q,PHI DOT',/,39X,'*',8X,'4',7X,'Q,P',/,1X,'AIRCRAFT ELEMENT ',
     2 'SPECIFICATION-',7X,'*',/,39X'*',/,2X,'ELEMENT  TYPE', 7X,
     3 'ELEMENT',10X,'*',/,2X,'NUMBER   CODE', 8X,'TYPE',12X,'*',/,3X,

     4 '(I)    (NTYPE(I))',20X,'*',/,39X,'*')

      DO 26 I=1,N
      K=NTYPE(I)
      IBEGIN=4*(K-1)+1
      IEND=4*(K-1)+4
      WRITE(W,27)I,NTYPE(I),(ELTYP(JTYP),JTYP=IBEGIN,IEND)
      FORMAT(3X,I2,6X,I2,5X,4A5,1X,'*')

      WRITE(W,28)
      FORMAT(/,1X,'CONTROL COLUMN DEFINITION (COLUMN C)-  B. NC(I,1)=',
     1 'NUMBER OF THE FIRST ELE-',/,39X,'*    MENT IN C ASSOCIATED WITH',
     2 ' AIR-',/,2X,'ELEMENT    (SEE INFOR-  (SEE INFOR-  *   CRAFT ',
     3 'ELEMENT I.',/,2X,'NUMBER',6X,'MATION-B)    MATION-C)    *',/,5X,
     4 'I',10X,'NC(I,1)',4X,'NC(I,2)',4X,'C. NC(I,2)=TOTAL NUMBER OF ',
     5 'C ELE-',/,39X,'*    MENTS FOR AIRCRAFT ELEMENT I.')

      DO 30 I=1,N
      WRITE(W,31)I,NC(I,1),NC(I,2)
      FORMAT(4X,I2,11X,I2,11X,I2,7X,'*')
      NCAND=LC+3
      NUMT=6
      WRITE(W,10)
      WRITE(W,32)NUMT,NCAND,NUMT,LC
      FORMAT(40X,'D. T HAS ',I1,' ELEMENTS, TAKEN FROM',/,1X,'DEFINIT',
     1 'ION OF THE TRIM ITERATION',4X,'*',3X,I2,' CANDIDATE ELEMENTS.',
     2 ' IT(I)',/,1X,'COLUMN T (SEE INFORMATION-D)-',9X,'*    NUMBERS',
     3 ' THE SPECIFIC ',I1,' OF THESE',/,39X,'*    TO BE USED IN T. THE',
     4 ' CANDIDATE',/,39X,'*    ELEMENTS COME FROM THE ',I2,' ROWS')
```

```
      WRITE(6,33)
      FORMAT(2X,'ELEMENT',11X,'VALUE',14X,'*    OF C, AND THE THREE ',
     1 'FLIGHT VAR-',/,4X,'IN',15X,'OF',14X,'*    IABLES THETA (PITCH ',
     2 'EULER',/,5X,'T',15X,'IT',16X,'*    ANGLE), PHI (ROLL EULER ',
     3 'ANGLE)',/,30X,'*    AND V (SIDESLIP VELOCITY).')

      DO 35 I=1,6
      WRITE(6,36)I,IT(I)
      FORMAT(5X,I1,15X,I2,16X,'*')

      NKEL=FOI
      NSELCT=LC-3
      IF(NSELCT)280,290,295
      CONTINUE
      IF(ISW.EQ.2) GO TO 483
      [illegible]

      FORMAT(/,1X,'CONTROL COLUMN DEFINITION (COLUMN C)-  B. NC(I,1)=',
     1 'NUMBER OF THE FIRST ELE-',/,30X,'*    MENT IN C ASSOCIATED WITH',
     2 ' AIR-',/,2X,'ELEMENT      (SEE INFOR-   (SEE INFOR-   *    CRAFT ',
     3 'ELEMENT I.',/,2X,'NUMBER',6X,'MATION-A)      MATION-C)     *',/,5X,
     4 'I',10X,'NC(I,1)',4X,'NC(I,2)',4X,'C. NC(I,2)=TOTAL NUMBER OF ',
     5 'C ELE-',/,30X,'*    MENTS FOR AIRCRAFT ELEMENT I.')

      DO 30 I=1,K
      WRITE(6,31)I,NC(I,1),NC(I,2)
      FORMAT(4X,I2,11X,I2,11X,I2,7X,'*')
      NCAND=LC+3
      NUMT=6
      WRITE(6,10)
      WRITE(6,37)NUKT,NCAND,NUMT,LC
      FORMAT(40X,'D. T HAS ',I1,' ELEMENTS, TAKEN FROM',/,1X,'DEFINIT',
     1 'ION OF THE THIK ITERATION',6X,'*',3X,I2,' CANDIDATE ELEMENTS, ',
     2 ' IT(I)',/,1X,'COLUMN T (SEE INFORMATION-D)-',9X,'*    NUMBERS',
     3 ' THE SPECIFIC ',I1,' OF THESE',/,30X,'*    TO BE USED IN T. THE',
     4 ' CANDIDATE',/,30X,'*    ELEMENTS COME FROM THE ',I2,' ROWS')

      WRITE(6,33)
      FORMAT(2X,'ELEMENT',11X,'VALUE',14X,'*    OF C, AND THE THREE ',
     1 'FLIGHT VAR-',/,4X,'IN',15X,'OF',14X,'*    IABLES THETA (PITCH ',
     2 'EULER',/,5X,'T',15X,'IT',16X,'*    ANGLE), PHI (ROLL EULER ',
     3 'ANGLE)',/,30X,'*    AND V (SIDESLIP VELOCITY).')

      DO 35 I=1,6
      WRITE(6,36)I,IT(I)
      FORMAT(5X,I1,15X,I2,16X,'*')
```

```
      NKEL=0
      NELNOT=LC-3
      IF(NELNOT)208,208,205
      CONTINUE
      IF(ISU.EQ.2) GO TO 603
      READ(5,120)(VNOT(J),J=1,NELNOT)
      CONTINUE
      WRITE(6,37)
      FORMAT(1H0,'VALUES FOR THOSE T-CANDIDATE ELEMENTS *',/,1X,
     1 'NOT SELECTED FOR T (SEE INFORMATION    *',/,1X,'ITEM D). ',
     2 'THESE VALUES ARE CONSTRAINTS *',/,1X,'ON THE TRIMMING ',
     3 'PROCESS.',14X,'*',/,39X,'*')

      DO 36 I=1,NELNOT
      WRITE(6,38) VNOT(I)
      FORMAT(1X,E15.4,23X,'*')

      CONTINUE

      DO 250 I=1,N


      NTYPI=NTYPE(I)
      GO TO(210,225,230,230),NTYPI

      LIFTING SURFACE INPUT/WRITE VERIFY

      CONTINUE
      IF(ISU.EQ.2) GO TO 604
      READ(5,100)(TITLE(K,I),K=1,20)
      CONTINUE
      WRITE(6,10)(TITLE(K,I),K=1,20)

      NPW(I)=20
      NINTS(I)=0

      WRITE(6,211)I
      FORMAT(/,1X,'VEHICLE ELEMENT NUMBER ',I2,/)

      IF(ISU.EQ.2) GO TO 605
      READ(5,300)(PW(J,I),J=1,19)
      CONTINUE
      FORMAT(5F12.3)
```

```
      WRITE(W,212)(PK(J,I),J=1,15)
      FORMAT(8X,'PSI L',8X,'THETA L',8X,'PHI L',10X,'AW',12X,'SW',/,
     1 5E14.4,//,8X,'CHORD',11X,'CD0',10X,'CD1',10X,'CD2',10X,'AWCLD',/,
     2 5E14.4,//,6X,'CAP GAMMA', 8X,'RW',10X,'LAMDA W',
     3 10X,'CM0',11X,'CMA',/,5E14.4,//)
      GO TO 250

      AERODYNAMIC BODY INPUT/WRITE VERIFY

      CONTINUE
      IF(ISU.EQ.2) GO TO 606
      READ(R,100)(TITLE(K,I),K=1,20)
      CONTINUE
      WRITE(W,10)(TITLE(K,I),K=1,20)

      NPK(I)=20
      NINTS(I)=0

      WRITE(W,226)I
      FORMAT(/,1X,'VEHICLE ELEMENT NUMBER ',I2,/)

      IF(ISU.EQ.2) GO TO 607
      READ(R,300)(PK(J,I),J=1,16)
      CONTINUE

      NPK(I)=20
      NINTS(I)=0

      WRITE(W,211)I
      FORMAT(/,1X,'VEHICLE ELEMENT NUMBER ',I2,/)

      IF(ISU.EQ.2) GO TO 608
      READ(R,300)(PK(J,I),J=1,15)
      CONTINUE
      FORMAT(5F10.0)

      WRITE(W,212)(PK(J,I),J=1,15)
      FORMAT(8X,'PSI L',8X,'THETA L',8X,'PHI L',10X,'AW',12X,'SW',/,
     1 5E14.4,//,8X,'CHORD',11X,'CD0',10X,'CD1',10X,'CD2',10X,'AWCLD',/,
     2 5E14.4,//,6X,'CAP GAMMA', 8X,'RW',10X,'LAMDA W',
     3 10X,'CM0',11X,'CMA',/,5E14.4,//)
      GO TO 250
```

```
 AERODYNAMIC BODY INPUT/WRITE VERIFY

 CONTINUE
 IF(ISU.EQ.2) GO TO 606
 READ(R,100)(TITLE(K,I),K=1,20)
 CONTINUE
 WRITE(W,10)(TITLE(K,I),K=1,20)

 NPK(I)=20
 NINTS(I)=0

 WRITE(W,226)I
 FORMAT(/,1X,'VEHICLE ELEMENT NUMBER ',I2,/)

 IF(ISU.EQ.2) GO TO 607
 READ(R,300)(PK(J,I),J=1,16)
 CONTINUE

 WRITE(W,227)(PK(J,I),J=1,16)
 FORMAT(8X,'PSI B',7X,'THETA B', 9X,'PHI B',10X,'AR',12X,'C0',/,
1 5E14.4,//,10X,'C1',12X,'C2',11X,'CY0',11X,'CY1',11X,'CZ0',/,
2 5E14.4,//, 9X,'CZ1',12X,'LH',11X,'CM0',11X,'CM1',11X,'CN0',/,
3 5E14.4,//, 9X,'CN1',/,E14.4,//)
 GO TO 250

 ROTOR INPUT/WRITE VERIFY

 CONTINUE
 IF(ISU.EQ.2) GO TO 608
 READ(R,100)(TITLE(K,I),K=1,20)
 CONTINUE
```

```
      WRITE(W,10)(TITLE(K,I),K=1,20)

      NPK(I)=200
      NINTS(I)=3

      WRITE(W,231)I
      FORMAT(/,1X,'VEHICLE ELEMENT NUMBER',I2,/)

      IF(ISU.EQ.2) GO TO 609
      READ(R,110)(INTS(J,I),J=1,3)
      CONTINUE
      WRITE(W,232)(INTS(J,I),J=1,3)
      FORMAT(1X,'NUMBER OF RADIAL STATIONS',I2,//,
     1 1X,'NUMBER OF AZIMUTHAL INTEGRATION ELEMENTS ',I2,//,
     2 1X,'NUMBER OF POINT MASSES ',I2,/)

      IF(ISU.EQ.2) GO TO 610
      READ(R,300)(PK(J,I),J=1,20)
      CONTINUE
      WRITE(W,233)(PK(J,I),J=1,20)
      FORMAT(7X,'OMEGA',11X,'RO',12X,'HA',12X,'RB',9X,'DELTA 0',/,
     1 5E14.4,//,5X,'DELTA 1',8X,'DELTA 2',7X,'DELTA 3',7X,'THETA 1',
     2 10X,'R',/,5E14.4,//,6X,'SMALL R',7X,'SMALL A',8X,'PSI R',8X,
     3 'THETA R',8X,'PHI R',/,5E14.4,//,2X,'BLADE NATURAL    BETA 0',
     4 8X,'BETA DOT 0',6X,'BETA',8X,'BETA DOT',/,5X,'FREQ. OVER',3X,
     5 '(ESTIMATED)    (ESTIMATED)  PERTURBATION  ',
     6 'PERTURBATION',/,6X,'OMEGA (P)', 6X,'(ROE)', 9X,'(RODE)', 9X,
     7 '(PB)', 9X,'(PBD)',/,5E14.4,/)

      WRITE(W,235)
      FORMAT(1X,'DISTRIBUTED BLADE PROPERTIES-',//,4X,'BLADE',42X,
     1 'FIRST',/,3X,'STATION   DISTRIBU-',5X, 'RADIAL',5X,'INITIAL',
     2 5X,'FLAPPING',/,4X,'NUMBER    TED MASS    DISTANCE',5X,'SHAPE',
     3 5X,'MODESHAPE',5X,'CHORD',/)

      NRS=INTS(1,I)
      DO 240 K=1,NRS
      ISTART=49+K
      IEND=129+K
      IF(ISU.EQ.2) GO TO 611
      READ(R,300)(PK(J,I),J=ISTART,IEND,20)
      CONTINUE
      WRITE(W,237)K,(PK(J,I),J=ISTART,IEND,20)
      FORMAT(5X,I2,3X,5E12.4)
      CONTINUE
```

```
      CONTINUE
      WRITE(W,232)(INTS(J,I),J=1,3)
      FORMAT(1X,'NUMBER OF RADIAL STATIONS',I2,//,
     1 1X,'NUMBER OF AZIMUTHAL INTEGRATION ELEMENTS ',I2,//,
     2 1X,'NUMBER OF POINT MASSES ',I2,/)

      IF(ISU.EQ.2) GO TO 610
      READ(R,300)(PK(J,I),J=1,20)
      CONTINUE
      WRITE(W,233)(PK(J,I),J=1,20)
      FORMAT(7X,'OMEGA',11X,'RO',12X,'BA',12X,'BB',9X,'DELTA 0',/,
     1 5E14.4,//,5X,'DELTA 1',8X,'DELTA 2',7X,'DELTA 3',7X,'THETA 1',
     2 10X,'R',/,5E14.4,//,6X,'SMALL B',7X,'SMALL A',8X,'PSI R',8X,
     3 'THETA R',8X,'PHI R',/,5E14.4,//,2X,'BLADE NATURAL    BETA 0',
     4 8X,'BETA DOT 0',6X,'BETA',8X,'BETA DOT',/,5X,'FREQ. OVER',3X,
     5 '(ESTIMATED)    (ESTIMATED)  PERTURBATION  ',
     6 'PERTURBATION',/,6X,'OMEGA (P)', 6X,'(BOE)', 9X,'(BDOE)', 9X,
     7 '(PB)', 9X,'(PBD)',/,5E14.4,/)

      WRITE(W,235)
      FORMAT(1X,'DISTRIBUTED BLADE PROPERTIES-',//,4X,'BLADE',42X,
     1 'FIRST',/,3X,'STATION   DISTRIBU-',5X, 'RADIAL',5X,'INITIAL',
     2 5X,'FLAPPING',/,4X,'NUMBER     TED MASS    DISTANCE',5X,'SHAPE',
     3 5X,'MODESHAPE',5X,'CHORD',/)

      NRS=INTS(1,I)
      DO 240 K=1,NRS
      ISTART=49+K
      IEND=129+K
      IF(ISU.EQ.2) GO TO 611
      READ(R,300)(PK(J,I),J=ISTART,IEND,20)
      CONTINUE
      WRITE(W,237)K,(PK(J,I),J=ISTART,IEND,20)
      FORMAT(5X,I2,3X,5E12.4)
      CONTINUE

      NPM=INTS(3,I)
      IF(NPM)245,245,244
      CONTINUE
      WRITE(W,242)
      FORMAT(/,1X,'POINT MASSES-',//,4X,'BLADE',42X,'FIRST',/,3X,
     1 'STATION',17X,'RADIAL',5X,'INITIAL',5X,'FLAPPING',/,4X,'NUMBER',
     2 6X,'MASS',6X,'DISTANCE',5X,'SHAPE',5X,'MODESHAPE',/)

      DO 248 K=1,NPM
      ISTART=149+K
      IEND=179+K
      IF(ISU.EQ.2) GO TO 612
      READ(R,300)(PK(J,I),J=ISTART,IEND,10)
      CONTINUE
```

```
      WRITE(W,237)K,(PK(J,I),J=ISTART,IEND,10)
      CONTINUE
      CONTINUE

      WRITE(W,249)
      FORMAT(//)

      CONTINUE

      IF(ISU.EQ.2) GO TO 613
      READ(R,130)(XEL(I),YEL(I),ZEL(I),I=1,N)
      READ(R,130)(((A(I,J,K),J=1,6),I=1,6),K=1,N)
      FORMAT(6F10.0)

      READ(R,140)(IX(I),JX(I),X(I),I=1,NEX)
      FORMAT(2(2I10,F10.0))


      NKEL=6*N

      READ(R,120)(TE(J),J=1,6)
      READ(R,120)(WF(J),J=1,NKEL)
      READ(R,120)WT,XCG,YCG,ZCG,TAS,RHO,PSIDOT,PTCHRT
      READ(R,120)ROLLRT,HDOT,((QINRTA(I,J),I=1,3),J=1,3)

      READ(R,120)PT,PV,PVDOT,PF,PTC,TACPT,WACPT,BACPT,BDACPT
      FORMAT(8F10.0)
      CONTINUE

      WRITE(W,255)
      FORMAT(1H1,'GEOMETRIC LOCATION OF THE ELEMENTS WITH RESPECT TO ',
     1 'THE OVERALL',/,1X,'VEHICLE AXIS SYSTEM-',//,4X,'ELEMENT',/,
     2 4X,'NUMBER',14X,'X',13X,'Y',13X,'Z',/)
      DO 257 I=1,N
      WRITE(W,258)I,XEL(I),YEL(I),ZEL(I)
      FORMAT(6X,I2,6X,3E14.4)

      WRITE(W,260)
      FORMAT(//1X,'INTERFERENCE VELOCITY CHARACTERISTIC AREA ',
     1 'MATRICES-',/)

      DO 265 I=1,N
      WRITE(W,262)(TITLE(K,I),K=1,20),I
      FORMAT(/,1X,18A4,/,1X,2A4,5X,'ELEMENT NUMBER',I2,/)
      WRITE(W,266)((A(J,K,I),K=1,6),J=1,6)
      FORMAT(1X,6F12.4)
```

```
      READ(R,130)(XEL(I),YEL(I),ZEL(I),I=1,N)
      READ(R,130)(((A(I,J,K),J=1,6),I=1,6),K=1,N)
      FORMAT(6F10.0)

      READ(R,140)(IX(I),JX(I),X(I),I=1,NEX)
      FORMAT(2(2I10,F10.0))


      NKEL=6*N

      READ(R,120)(TE(J),J=1,6)
      READ(R,120)(WE(J),J=1,NKEL)
      READ(R,120)WT,XCG,YCG,ZCG,TAS,RHO,PSIDOT,PTCHRT
      READ(R,120)ROLLRT,HDOT,((QINRTA(I,J),I=1,3),J=1,3)

      READ(R,120)PT,PV,PVDOT,PF,PTC,TACPT,WACPT,HACPT,BDACPT
      FORMAT(8F10.0)
      CONTINUE

      WRITE(W,255)
      FORMAT(1H1,'GEOMETRIC LOCATION OF THE ELEMENTS WITH RESPECT TO ',
     1 'THE OVERALL',/,1X,'VEHICLE AXIS SYSTEM-',//,4X,'ELEMENT',/,
     2 4X,'NUMBER',14X,'X',13X,'Y',13X,'Z',/)
      DO 257 I=1,N
      WRITE(W,258)I,XEL(I),YEL(I),ZEL(I)
      FORMAT(6X,I2,6X,3F14.4)

      WRITE(W,260)
      FORMAT(//1X,'INTERFERENCE VELOCITY CHARACTERISTIC AREA ',
     1 'MATRICES-',/)

      DO 265 I=1,N
      WRITE(W,262)(TITLE(K,I),K=1,20),I
      FORMAT(/,1X,18A4,/,1X,2A4,5X,'ELEMENT NUMBER',I2,/)
      WRITE(W,266)((A(J,K,I),K=1,6),J=1,6)
      FORMAT(1X,6F12.4)
      WRITE(W,268)(IX(J),JX(J),X(J),J=1,NEX)
      FORMAT( 1H1,'INTERFERENCE VELOCITY COUPLING MATRIX,X-',//,10X,
     1 'I',11X,'J',6X,'X(I,J)',//,(9X,I2,10X,I2,E12.4))

      WRITE(W,270)(J,TE(J),J=1,6)
      FORMAT(/,1X,'ESTIMATED TWIN COLUMN (TE)-',//,14X,'ROW NUMBER (I)',
     1 13X,'TE(I)',/,(17X,I2,16X,F12.4))

      WRITE(W,272)
      FORMAT(/,1X,'ESTIMATED INTERFERENCE VELOCITY COLUMN (WE)-',//,
     1 1X,'VEHICLE',/,1X,'ELEMENT',/,2X,'NUMBER',4X,'UW', 9X,'VW', 9X,
     2 'WW', 9X,'PW', 9X,'QW', 9X,'RW',/)

      DO 274 I=1,N
```

NOT REPRODUCIBLE

```
      ISTART=6*(I-1)+1
      IEND=ISTART+5
      WRITE(W,275)I,(WE(J),J=ISTART,IEND)
      FORMAT(2X,I2,2X,6E11.4)

      WRITE(W,277)
      FORMAT( 1H1,'TRIM PROBLEM DEFINITION-',//,7X,'GROSS',51X,
     1 'INERTIAL',/,7X,'WEIGHT',10X,'XCG',11X,'YCG',11X,'ZCG',10X,
     2 'SPEED')

      WRITE(W,278)WT,XCG,YCG,ZCG,TAS,RHO,PSIDOT,PTCHRT,ROLLRT,HDOT
      FORMAT(1X,5E14.4,//,9X,'RHO', 8X,'PSI DOT',7X,'PTCHRT',8X,
     1 'ROLLRT', 9X,'H DOT',/,5E14.4,/)

      WRITE(W,279)((QINRTA(J,K),K=1,3),J=1,3)
      FORMAT( 9X,'IXX',11X,'IXY',11X,'IXZ',11X,'IYX',11X,'IYY',
     1 /,5E14.4,//, 9X,'IYZ',11X,'IZX',11X,'IZY',11X,'IZZ',/,4E14.4,/)

      WRITE(W,280)PT,PV,PVDOT,PF,PTC,TACPT,WACPT,BACPT,BDACPT
      FORMAT(1X,'MISCELLANEOUS ITEMS-',//,7X,'PERTURB',7X,'PERTURB',7X,
     1 'PERTURB',7X,'PERTURB',7X,'PERTURB',/,10X,'T',13X,'V',11X,
     2 'V DOT',11X,'F',13X,'C',/5E14.4,//,8X,'ACCEPT',8X,'ACCEPT',8X,
     3 'ACCEPT',8X,'ACCEPT',/,10X,'T',13X,'W',12X,'BETA',7X,'BETA DOT',
     4 /,4E14.4,/)

      GO TO 999
      CONTINUE

      WRITE(W,310)
      FORMAT(1H0,1X,'ESTIMATED AND COMPUTED VARIABLE COLUMNS-',/,2X,
     1 'ROW',/,2X,'NO.',6X,'CO', 9X,'TE', 9X,'SO', 9X,'RO', 9X,'P ',
     2  9X,'DT',/)

      IEND=6
      IF(LC.GT.6)IEND=LC
      DO 315 I=1,IEND
      IF(I.GT.6)GO TO 313
      WRITE(W,314)I,CO(I),TE(I),SO(I),RO(I),P(I),DT(I)
      FORMAT(2X,I2,2X,6E11.4)
      GO TO 315
      WRITE(W,314)I,CO(I)
      CONTINUE

      WRITE(W,316)
      FORMAT(/,1X,'VEHICLE',/,1X,'ELEMENT',/,2X,'NUMBER',3X,'BW', 9X,
     1 'WF', 9X,'WB', 8X,'VIB',8X,'VAB', 9X,'FB',/)
```

```
WRITE(W,270)RHO,PSIDOT,PTCHRT,ROLLRT,HDOT
FORMAT(1X,5E14.4,//,9X,'RHO', 8X,'PSI DOT',7X,'PTCHRT',8X,
1 'ROLLRT', 9X,'H DOT',/,5E14.4,/)

WRITE(W,279)((QINRTA(J,K),K=1,3),J=1,3)
FORMAT( 9X,'IXX',11X,'IXY',11X,'IXZ',11X,'IYX',11X,'IYY',
1 /,5E14.4,//, 9X,'IYZ',11X,'IZX',11X,'IZY',11X,'IZZ',/,4E14.4,/)

WRITE(W,280)PT,PV,PVDOT,PF,PTC,TACPT,WACPT,BACPT,BDACPT
FORMAT(1X,'MISCELLANEOUS ITEMS-',//,7X,'PERTURB',7X,'PERTURB',7X,
1 'PERTURB',7X,'PERTURB',7X,'PERTURB',/,10X,'T',13X,'V',11X,
2 'V DOT',11X,'F',13X,'C',/5E14.4,//,8X,'ACCEPT',8X,'ACCEPT',8X,
3 'ACCEPT',8X,'ACCEPT',/,10X,'T',13X,'W',12X,'BETA',7X,'BETA DOT',
4 /,4E14.4,/)

GO TO 999
CONTINUE

WRITE(W,310)
FORMAT(1H0,1X,'ESTIMATED AND COMPUTED VARIABLE COLUMNS-',/,2X,
1 'ROW',/,2X,'NO.',6X,'CO', 9X,'TE', 9X,'SO', 9X,'RO', 9X,'P ',
2  9X,'DT',/)

IEND=6
IF(LC.GT.6)IEND=LC
DO 315 I=1,IEND
IF(I.GT.6)GO TO 313
WRITE(W,314)I,CO(I),TE(I),SO(I),RO(I),P(I),DT(I)
FORMAT(2X,I2,2X,6F11.4)
GO TO 315
WRITE(W,314)I,CO(I)
CONTINUE

WRITE(W,316)
FORMAT(/,1X,'VEHICLE',/,1X,'ELEMENT',/,2X,'NUMBER',3X,'DW', 9X,
1 'WE', 9X,'WO', 8X,'VIO',8X,'VAO', 9X,'FO',/)

DO 320 I=1,M
IST=6*(I-1)+1
IED=IST+5
WRITE(W,318)I,(DW(J),WE(J),WO(J),VIO(J),VAO(J),FO(J),J=IST,IED)
FORMAT(3X,  I2,/,(6X,6E11.4))

WRITE(W,322)
FORMAT(/,1X,'ROTOR BLADE MOTION HISTORIES-',/)

DO 326 I=1,N
IF(ITYP(I).NE.3)GO TO 328
[illegible]
[illegible]
```

```
      DELPSI=360.0/QNAS
      NAP=NAS+1

      WRITE(W,323)(TITLE(K,I),K=1,20)
      FORMAT(/,1X,18A4,/,1X,2A4,/,21X,'PSI',10X,'BETA',9X,'BETA DOT',/,
     1 18X,'(DEGREES)',4X,'(RADIANS)',6X,'(RAD/SEC)',/)

      DO 327 J=1,NAP
      QJAY=J
      PSIDEG=DELPSI*(QJAY-1.0)
      WRITE(W,324)PSIDEG,BETA(J,I),BETADT(J,I)
      FORMAT(10X,F15.2,2E15.4)

      CONTINUE

      WRITE(W,330)NPASS
      FORMAT(//1X,'NUMBER OF ITERATION CYCLES ',I2,/)

      GO TO 999
      CONTINUE

      WRITE(W,355)
      FORMAT(1H1,'CONTRIBUTIONS BY THE INDIVIDUAL VEHICLE COMPONENTS ',
     1 'TO THE FVA MATRIX.',/,3X,'(FVA IS THE AERODYNAMIC VELOCITY ',
     2 'FORCING MATRIX)'/)

      DO 362 I=1,M
      IST=6*(I-1)+1
      IND=IST+5
      IF(I.NE.5.AND.I.NE.9) GO TO 357
      WRITE(W,355)
      CONTINUE

      WRITE(W,358)(TITLE(K,I),K=1,20),I
      FORMAT(/,3X,17A4,/,3X,3A4,10X,'(VEHICLE ELEMENT NUMBER',I2,')',/,
     1 12X,'UA', 9X,'VA', 9X,'WA', 9X,'PA', 9X,'QA', 9X,'RA')

      DO 360 J=IST,IND
      LBLS=J-IST+1
      WRITE(W,361)ROWS(LBLS),(FVA(J,K),K=IST,IND)
      FORMAT(5X,A1,6E11.4)
      CONTINUE

      WRITE(W,365)
      FORMAT(1H1,'CONTRIBUTIONS BY THE INDIVIDUAL VEHICLE COMPONENTS ',
     1 'TO THE FVI MATRIX.',/,3X,'(FVI IS THE INERTIAL VELOCITY ',
```

```
WRITE(6,324)PSIDEG,BETA(J,I),BETADT(J,I)
FORMAT(10X,F15.2,2E15.4)

CONTINUE

WRITE(6,330)NPASS
FORMAT(//3X,'NUMBER OF ITERATION CYCLES ',I2,/)

GO TO 999
CONTINUE

WRITE(6,355)
FORMAT(1H1,'CONTRIBUTIONS BY THE INDIVIDUAL VEHICLE COMPONENTS ',
1 'TO THE FVA MATRIX.',/,3X,'(FVA IS THE AERODYNAMIC VELOCITY ',
2 'FORCING MATRIX)'/)

DO 302 I=1,N
IST=6*(I-1)+1
IND=IST+5
IF(I.NE.5.AND.I.NE.9) GO TO 357
WRITE(6,355)
CONTINUE

WRITE(6,358)(TITLE(K,I),K=1,20),I
FORMAT(/,3X,17A4,/,3X,3A4,10X,'(VEHICLE ELEMENT NUMBER',I2,')',/,
1 12X,'LA', 9X,'VA', 9X,'WA', 9X,'PA', 9X,'QA', 9X,'RA')

DO 360 J=IST,IND
LBLS=J-IST+1
WRITE(6,341)ROWS(LBLS),(FVA(J,K),K=IST,IND)
FORMAT(5X,A1,6E11.4)
CONTINUE

WRITE(6,345)
FORMAT(1H1,'CONTRIBUTIONS BY THE INDIVIDUAL VEHICLE COMPONENTS ',
1 'TO THE FVI MATRIX.',/,3X,'(FVI IS THE INERTIAL VELOCITY ',
2 'FORCING MATRIX)')

DO 372 I=1,N
IST=6*(I-1)+1
IND=IST+5
IF(I.NE.5.AND.I.NE.9) GO TO 367
WRITE(6,355)
CONTINUE

WRITE(6,344)(TITLE(K,I),K=1,20),I
FORMAT(/,3X,17A4,/,3X,3A4,10X,'(VEHICLE ELEMENT NUMBER',I2,')',/,
1 12X,'LI', 9X,'VI', 9X,'WI', 9X,'PI', 9X,'QI', 9X,'RI')
DO 371 J=IST,IND
LBLS=J-IST+1
```

```
      WRITE(6,381)IROWS(LBLS),(FV1(J,K),K=IST,IND)
      CONTINUE

      WRITE(6,375)
      FORMAT(1H1,'CONTRIBUTIONS BY THE INDIVIDUAL VEHICLE COMPONENTS ',
     1 'TO THE  CONTROL',/,1X,'FORCING COLUMN FC.',/)

      ICOUT=0
      DO 377 I=1,N
      NC1=NC(I,2)
      IF(NC(I,2))382,382,376
      IF(ICOUT.NE.5.AND.ICOUT.NE.0) GO TO 377
      WRITE(6,375)
      CONTINUE

      ICOUT=ICOUT+1
      WRITE(6,378)(TITLE(K,1),K=1,20),I,(CLABEL(L),L=1,NC1)
      FORMAT(/,3X,17A4,/,3X,3A4,10X,'(VEHICLE ELEMENT NUMBER',I2,')',/,
     1 11X,6(/5,6X))
      IST=NC(I,1)
      IND=IST+NC(I,2)-1
      WRITE(6,379)(K,K=IST,IND)
      FORMAT(1H0,17X,6(I2,9X))
      INOST=6*(I-1)+1
      INCR=INOST+5
      DO 378 J=INOST,INCR
      LBLS=J-INOST+1
      WRITE(6,381)IROWS(LBLS),(FC(J,K),K=IST,IND)
      FORMAT(5X,A1,6E11.4)
      CONTINUE

      GO TO 900
      CONTINUE

      DO 400 LP=1,2

      WRITE(6,401)(COLS(J,LP),J=1,6)
      FORMAT(///,6X,6(6X,A5))

      DO 403 J=1,6

      WRITE(6,381)IROWS(J),(STD(J,K,LP),K=1,6)
      CONTINUE
      IC=0
      CONTINUE
      IST=IF+N+1
```

```
      IF(NC(I,2))300,300,376
      IF(ICOUT.NE.5.AND.ICOUT.NE.9) GO TO 377
      WRITE(6,375)
      CONTINUE

      ICOUT=ICOUT+1
      WRITE(6,378)(TITLE(K,I),K=1,20),I,(CLABEL(L),L=1,NC(I))
      FORMAT(/,3X,17A4,/,3X,3A4,18X,'(VEHICLE ELEMENT NUMBER',I2,')',/,
     1 11X,6(A5,6X))
      ISTART(I,1)
      I[illegible]=ISTART(I,2)-1
      WRITE(6,379)(X,N=IST,IND)
      FORMAT(1H0,12X,6(I2,9X))
      I[illegible](I-1)+1
      I[illegible]=I[illegible]+5
      DO 395 J=I[illegible],I[illegible]
      L[illegible]=J-I[illegible]+1
      WRITE(6,39)[illegible](L,3),(PC(J,K),K=IST,IND)
      FORMAT(1X,A1,6F11.4)
      CONTINUE

      GO TO 300
      CONTINUE

      DO 400 L=1,2

      WRITE(6,401)(COL[illegible](J,L),J=1,6)
      FORMAT(///,4X,6(6X,A5))

      DO 403 J=1,6

      WRITE(6,30)[illegible](J),(STD(J,K,L),K=1,6)
      CONTINUE
      I[illegible]
      CONTINUE
      IST=I[illegible]
      I[illegible]=IST+5
      IF(I[illegible].GT.LC) I[illegible]

      [illegible]=I[illegible]-IST+1

      WRITE(6,400)(CLABEL(L),L=1,[illegible])
      FORMAT(///,11X,6(A5,6X))

      WRITE(6,479)(K,K=IST,IEND)
      FORMAT(1H0,12X,6(I2,9X))

      DO 412 J=1,6
      WRITE(6,30)[illegible](J),(PC(J,K),K=IST,IEND)
```

```
      IF(IEND.LT.LC) GO TO 407
      GO TO [illegible]
      CONTINUE
      RETURN
      END

      SUBROUTINE SWEEP(ISWEEP,VAS,VIS,VIDOTS,A,FM,[illegible])

      THREE OPTIONS ARE AVAILABLE IN SWEEP, AS KEYED BY (ISWEEP)

         ISWEEP                        OPTION

           1      FLEXIBLE BLADES- NO HUB LOAD REQUIRED.
           2      FLEXIBLE BLADES- HUB LOADS REQUIRED.
           3      RIGID BLADES- HUB LOADS REQUIRED.

      DIMENSION VAS(6),VIS(6),VIDOTS(6),A(3),FM(6),FA(6)

      COMMON/[illegible]/ [illegible]
     1 [illegible]
     2 [illegible]
     3 [illegible]
     4 [illegible]
     5 [illegible]
      COMMON/[illegible]
      COMMON/[illegible]

      PERMANENT DEBUG

      IF([illegible].EQ.1) GO TO 20
      WRITE(6,[illegible])
      FORMAT(1X,'SWEEP HAS BEEN CALLED',[illegible])
      CONTINUE

      END PERMANENT DEBUG

      COMPUTE CONSTANTS AND INITIALIZE VARIABLES OF INTEGRATION.

      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      DO 2 I=1,6
      FM(I)=0.0

      BEGIN SWEEPING THE ROTOR DISC, BEGINNING WITH THE AZIMUTHAL LOOP.
```

```
C        2     FLEXIBLE BLADES- HUB LOADS REQUIRED.
C        3     RIGID BLADES- HUB LOADS REQUIRED.
C
      DIMENSION VAS(6),VIS(6),VIDOTS(6),A(3),FH(6),FA(6)
C
      COMMON/PHYSCS/ OMEG,CAPRO,CAPRA,CAPRB,DELT0,DELT1,DELT2,DELT3,
     1 THET1,[illegible]
     2 [illegible]
     3 [illegible]
     4 [illegible]
     5 [illegible]
      COMMON/[illegible]
      COMMON/[illegible]/IBMIST,BETA(30,8),BETADT(30,8)
C
C     PERMANENT DEBUG
C
      IF(NDBG.EQ.1) GO TO 20
      WRITE(6,[illegible]) [illegible]
    [illegible] FORMAT(1X,'SWEEP HAS BEEN CALLED',[illegible])
    [illegible] CONTINUE
C
C     END PERMANENT DEBUG
C
C
C     COMPUTE CONSTANTS AND INITIALIZE VARIABLES OF INTEGRATION.
C
      SCV=0.0
      CCV=1.0
      [illegible]
      [illegible]
      [illegible]
      DO 7 I=1,6
    7 FH(I)=0.0
C
C     BEGIN SWEEPING THE ROTOR DISC, BEGINNING WITH THE AZIMUTHAL LOOP.
C
      DO 930 J=1,NAS
C
C     CALCULATE QUANTITIES CONSTANT TO THE RADIAL INTEGRATION,AND INIT-
C     IALIZE INTEGRAL VARIABLES.
C
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
```

```
      FA(NITL)=0.0
      PXAC=0.0
      PYAC=0.0
      PZAC=0.0
      ZDD=0.0
      SDD=0.0
      IDD=0
      ZIDP=BLDPRP(1,4)
C
C     CALCULATE THE BLADE TIP DISTRIBUTED AIRLOAD.
C
      K=NS
C
C     CALCULATE BLADE ELEMENT POSITION,VELOCITY, SLOPE AND INCIDENCE.
C
      S=BLDPRP(K,2)
      DS=BLDPRP(K,2)-BLDPRP(K-1,2)
      Z=BLDPRP(K,3)+BLDPRP(K,4)*Q
      ZD=BLDPRP(K,4)*QD
      ZP=((BLDPRP(K,4)-BLDPRP(K-1,4))*Q+BLDPRP(K,3)-BLDPRP(K-1,3))/DS
      THETA=THETT+(THETI*BLDPRP(K,2))/R
      CHORD=BLDPRP(K,5)
C
C     ASSEMBLE BLADE-ELEMENT CO-ORDINATE TRANSFORMATION MATRIX.
C
      T11=1.0
      T12=0.0
      T13=-ZP
      T21=-THETA*ZP
      T22=1.0-0.5*THETA**2
      T23=-THETA
      T31=ZP
      T32=THETA
      T33=T22
C
C     CALCULATE ELEMENT LOCAL AIRSPEED IN ROTATING HUB CO-ORDINATES.
C
      UXT=VIS(1)*CSV-VIS(2)*SNV
      UYT=VIS(2)*CSV+VIS(1)*SNV
      PXT=VIS(4)*CSV+VIS(5)*SNV
      PYT=VIS(5)*CSV-VIS(4)*SNV
      UROT=S*OMEGA
      VROT=-S*(VIS(6)-OMEG)-Z*PYT
      W=VIS(3)-S*PXT-ZD
C
C     ROTATE THESE VELOCITY COMPONENTS TO BLADE COORDINATES USING T.
```

```
K=NBS

CALCULATE BLADE ELEMENT POSITION,VELOCITY, SLOPE AND INCIDENCE.

S=BLDPRP(K,2)
DS=BLDPRP(K,2)-BLDPRP(K-1,2)
Z=BLDPRP(K,3)+BLDPRP(K,4)*S
ZO=BLDPRP(K,4)*RO
ZP=((BLDPRP(K,4)-BLDPRP(K-1,4))*S+BLDPRP(K,3)-BLDPRP(K-1,3))/DS
THETA=THETTS+(THETT*BLDPRP(K,2))/R
CONJ=BLDPRP(K,5)

ASSEMBLE BLADE-ELEMENT CO-ORDINATE TRANSFORMATION MATRIX.

T11=1.0
T12=0.0
T13=-ZP
T21=-THETA*ZP
T22=1.0-0.5*THETA**2
T23=-THETA
T31=ZP
T32=THETA
T33=T22

CALCULATE ELEMENT LOCAL AIRSPEED IN ROTATING HUB CO-ORDINATES.

U1T=VIS(1)*CCV-VIS(2)*SCV
U2T=VIS(2)*CCV+VIS(1)*SCV
P1T=VIS(4)*CCV+VIS(5)*SCV
P2T=VIS(4)*CCV-VIS(5)*SCV
UW=U1T+Z*P2T
VW=U2T-R*(VIS(6)-OMEG(6))-Z*P1T
WW=VIS(3)+S*P1T+ZO

ROTATE THESE VELOCITY COMPONENTS TO CHORD COORDINATES USING T.

UC=T21*UW+T22*VW+T23*WW
UA=T31*UW+T32*VW+T33*WW

COMPUTE THE TIP LOSS FACTOR

[illegible]
[illegible]
[illegible]
[illegible]

PERFORM RADIAL INTEGRATION.

DO 310 K=2,NBS
```

```
      CALCULATE BLADE ELEMENT POSITION,VELOCITY, SLOPE AND INCIDENCE.

      S=BLDPRP(K,7)
      DS=BLDPRP(K,2)-BLDPRP(K-1,2)
      Z=BLDPRP(K,3)+BLDPRP(K,4)*Q
      ZD=BLDPRP(K,4)*QD
      ZP=((BLDPRP(K,4)-BLDPRP(K-1,4))*Q+BLDPRP(K,3)-BLDPRP(K-1,3))/DS
      THETA=THETP+(THETI*BLDPRP(K,2))/R
      ZI=BLDPRP(K,4)
      CHRD=BLDPRP(K,5)
      IF(S.LE.RF) GO TO 175
      IFOLD=1
      S=RF
      SCALE=(S-SOR)/DS
      [illegible]
      Z=ZOD+SCALE*(Z-ZOD)
      ZD=ZDOD+SCALE*(ZD-ZDOD)
      ZP=ZPOD+SCALE*(ZP-ZPOD)
      THETA=THTAOD+SCALE*(THETA-THTAOD)
      ZI=ZIOD+SCALE*(ZI-ZIOD)
      CHRD=CHRDOD+SCALE*(CHRD-CHRDOD)
      CONTINUE

      ASSEMBLE BLADE-ELEMENT CO-ORDINATE TRANSFORMATION MATRIX.

      T11=1.0
      T12=0.0
      T13=-ZP
      T21=-THETA*ZP
      T22=1.0-0.5*THETA**2
      T23=-THETA
      T31=ZP
      T32=THETA
      T33=T22

      CALCULATE ELEMENT LOCAL AIRSPEED IN ROTATING HUB CO-ORDINATES.

      [illegible]
      [illegible]
      [illegible]

      ROTATE THESE VELOCITY COMPONENTS TO CHORD CO-ORDINATES USING T.

      UC=T11*UH+T21*VH+T31*WH
      WC=T13*UH+T23*VH+T33*WH
```

```
      C[illegible](K,S)
      IF(S.LE.RS) GO TO 175
      IPONE=1
      SO=S
      SCALE=(S-SO)/DS
      RS=S-SO
      Z=ZO+SCALE*(Z-ZO)
      ZP=ZPO+SCALE*(ZP-ZPO)
      ZPP=ZPPO+SCALE*(ZPP-ZPPO)
      THETA=THETAO+SCALE*(THETA-THETAO)
      Z1=Z1O+SCALE*(Z1-Z1O)
      CORD=CORDO+SCALE*(CORD-CORDO)
      CONTINUE

C     ASSEMBLE BLADE-ELEMENT CO-ORDINATE TRANSFORMATION MATRIX.

      T11=1.0
      T12=0.0
      T13=-ZP
      T21=-THETA*ZP
      T22=1.0-0.5*THETA**2
      T23=-THETA
      T31=ZP
      T32=THETA
      T33=1.0

C     CALCULATE ELEMENT LOCAL AIRFLOW IN ROTATING HUB CO-ORDINATES.

      UR=U1*Z*P1
      VR=[illegible]-S*(VAS(6)-[illegible](6))-Z*P
      WR=VAS(3)+S*P1*ZP

C     ROTATE THESE VELOCITY COMPONENTS TO CHORD CO-ORDINATES USING T.

      UC=T11*UR+T12*VR+T13*WR
      UN=T31*UR+T32*VR+T33*WR

C     CALCULATE CHORD AND NORMAL-TO-CHORD FORCE DISTRIBUTION FUNCTIONS.

      FC=[illegible]
      FN=[illegible]

C     COMPUTE BLADE AERODYNAMIC DISTRIBUTED FORCE FUNCTIONS AT THE POINT
C     S IN ROTATING HUB CO-ORDINATES BY ROTATING FCB AND FNB USING THE
C     TRANSPOSE OF T.

      PXA=T11*FC+T31*FN
      PYA=T12*FC+T32*FN
      PZA=T13*FC+T33*FN
```

ADD THE CONTRIBUTION OF THIS ELEMENT TO THE FORCES.

```
FG=FG+0.5*(Z1*PZA+Z1OD*PZAOD)*DS
IF(IPRESW.EQ.1) GO TO 290
FA(1)=FA(1)+0.5*(PXA+PXAOD)*DS
FA(2)=FA(2)+0.5*(PYA+PYAOD)*DS
FA(3)=FA(3)+0.5*(PZA+PZAOD)*DS
FA(4)=FA(4)-0.5*(Z*PYA+ZOD*PYAOD)*DS
FA(5)=FA(5)+0.5*(Z*PXA+S*PZA+ZOD*PXAOD+SOD*PZAOD)*DS
FA(6)=FA(6)+0.5*(-S*PYA-SOD*PYAOD)*DS
CONTINUE
IF(IFCAL.EQ.1) GO TO 301
PXAOD=PXA
PYAOD=PYA
PZAOD=PZA
ZOD=Z
SOD=S
Z1OD=Z1
ZPOD=ZP
ZOOD=ZO
THTAOD=THETA
COMOD=COMO
CONTINUE
CONTINUE
```

DEBUG

```
IF(NDBUG.LT.1) GO TO 80
WRITE(6,75)
WRITE(6,75)S,DS,Z,ZO,ZP,THETA,Z1,COMO,P,BO,FG,(FA(N),N=1,6)
FORMAT(1X,11E11.4)
CONTINUE
```

END DEBUG

ALTER THE FORCES BY THE NECESSARY CONSTANTS.

```
FG=-FG*WFC2
IF(IFCASE.EQ.1)GO TO 322
DO 320 N=1,6
FA(N)=-FA(N)*RHOO2
CONTINUE
```

ADD INERTIAL TERMS TO THE FORCES.

```
CONTINUE
IF(IDONE.EQ.1) GO TO 301
PXAOD=PXA
PYAOD=PYA
PZAOD=PZA
ZOD=Z
SOD=S
Z1OD=Z1
ZPOD=ZP
ZDOD=ZD
THTAOD=THETA
CORDOD=CORD
CONTINUE
CONTINUE

DEBUG

IF(NOXS.EQ.1) GO TO 80
WRITE(6,75)
WRITE(6,75)S,DS,Z,ZD,ZP,THETA,Z1,CORD,P,BO,FG,(FA(M),M=1,6)
FORMAT(1X,11E11.4)
CONTINUE

END DEBUG


ALTER THE FORCES BY THE NECESSARY CONSTANTS.

FG=-FG*RHO02
IF(INCXSP.EQ.1)GO TO 322
DO 320 M=1,6
FA(M)=-FA(M)*RHO02
CONTINUE

ADD INERTIAL TERMS TO THE FORCES.

FG=FG-BMI1*VIDOTS(3)-BMI2*RTETRM
IF(INCXSP.EQ.1) GO TO 390
FA(3)=FA(3)-BMI1*(FG/BMB-ROMEG2*R)
FA(5)=FA(5)-BMI2*((OMEG2-SOMG2)*R+FG/BMB)+PETERM

RESOLVE ROTATING HUB LOADS TO THE NON-ROTATING HUB CO-ORDINATE
SYSTEM.

FH(1)=FH(1)+FA(1)*CCV+FA(2)*SCV
FH(2)=FH(2)+FA(2)*CCV-FA(1)*SCV
FH(3)=FH(3)+FA(3)
FH(4)=FH(4)+FA(4)*CCV+FA(5)*SCV
FH(5)=FH(5)+FA(5)*CCV-FA(4)*SCV
FH(6)=FH(6)+FA(6)
```

```
      CONTINUE
C
C     ADVANCE B,BD,SCY AND CCY TO THE NEXT AZIMUTH POSITION.
C
      IF(INDXSP.EQ.3) GO TO 400
      BETA(J,IOHIST)=B
      BETADT(J,IOHIST)=BD
      FACT=B-FG/(SOMG2*OMG)
      B=FACT*COT+(BD/SOMG)*SOT+B-FACT
      BD=-FACT*SOMG*SOT+BD*COT
      CONTINUE
C
      SCYO=SCY
      SCY=SCY*CDPSI+CCY*SDPSI
      CCY=CCY*CDPSI-SCYO*SDPSI
      CONTINUE
C
C     MULTIPLY HUB LOADS BY CORRECT CONSTANTS.
C
      IF(INDXSP.EQ.1) GO TO 950
      SROR=SR/GNAR
      DO 905 M=1,6
      FH(M)=FH(M)*SROR
      CONTINUE
      BETA(NAS+1,IOHIST)=B
      BETADT(NAS+1,IOHIST)=BD
C
C     PERMANENT DEBUG
C
      IF(NCXS.EQ.1) GO TO 955
      WRITE(6,954)B,BD,(FH(M),M=1,6)
      FORMAT(1X,'SWEEP HAS BEEN EXECUTED',2E20.8,6E10.3)
      CONTINUE
C
C     END PERMANENT DEBUG
C
      RETURN
      END
```

```
      SUBROUTINE BODY(I,VA,FB)

      DIMENSION VA(48),C(12),FB(48)

      COMMON/PHYSCS/ER1,ER2,ER3,CA0,C0,C1,C2,CY0,CY1,CZ0,CZ1,CLB,CM0,
     1 CM1,CN0,CN1,F
      COMMON/SPLC/WT,XCG,YCG,ZCG,TAS,RHO,PSIDOT,PTCHRT,ROLLRT,HDOT,LC,
     1 NCPIAR,IT(4),GIMBTA(3,3)
      COMMON/DHAT B/BV1(6),BV2(6),BV3(6)
      COMMON/INDECS/NC(8,7),NTYPE(8),NTHRU(8),N,NPASS, NDIRCT,NEX,NITER

      NTHRU(I)=NTHRU(I)+1
      IF(NTHRU(I).GT.1) GO TO 5

      COMPUTE CONSTANT TERMS AND FACTORS USED BY BODY EQUATIONS.

      F=0.5*RHO*TAS
    5 CONTINUE

      COMPUTE COMPONENTS OF LOCAL WIND IN BODY CO-ORDINATES.

      NST=6*(I-1)+1
      CALL EULER(NST,1,2,1,48,6,ER1,ER2,ER3,VA,BV1)

      CALCULATE LOADS GENERATED BY BODY IN BODY CO-ORDINATES.

      BV2(1)=-F*BV1(1)*(BV1(1)*CA0+BV1(3)*C1+BV1(2)*C2)
      BV2(2)=-F*BV1(1)*(BV1(2)*CY1+BV1(1)*CY0)
      BV2(3)=-F*BV1(1)*(BV1(3)*CZ1+BV1(1)*CZ0)
      BV2(4)=0.0
      BV2(5)=F*CLB*BV1(1)*(BV1(3)*CM1+BV1(1)*CM0)
      BV2(6)=F*CLB*BV1(1)*(BV1(2)*CN1+BV1(1)*CN0)

      ROTATE BODY LOADS BACK TO AIRPLANE CO-ORDINATES.

      CALL EULER(1,NST,2,7,6,48,ER1,ER2,ER3,BV2,FB)
      RETURN
      END
```

```
      SUBROUTINE LIFT(I,VA,C,FB)

      DIMENSION VA(40),C(12),FB(60)

      COMMON/[illegible]/EL1,EL2,EL3,AV,SV,CHORD,CR0,CR1,CR2,ANGLE,[illegible],
     1 [illegible],[illegible],[illegible],[illegible],DAMP,PITCH,YAW

      COMMON/[illegible]/EV1(6),EV2(6),EV3(6)
      COMMON/[illegible]/X(8,7),NTYPE(8),[illegible](8),N,NPASS,[illegible],NEX,[illegible]
      COMMON/[illegible]/DT,XCG,YCG,ZCG,TAS,[illegible],PSIDOT,PTCHDT,ROLLDT,[illegible],LC,
     1 [illegible],IT(6),[illegible](3,9)

      NTURN(I)=NTURN(I)+1
      IF(NTURN(I).GT.1) GO TO 5

C     COMPUTE CONSTANT TERMS AND FACTORS USED BY LIFT EQUATIONS.

      [illegible]
      [illegible]((1.0+2.0*[illegible])/(6.0*(1.0+[illegible])))
      [illegible]((1.0+3.0*[illegible])/(24.0*(1.0+[illegible])))
      [illegible]
      [illegible]
    5 CONTINUE

C     COMPUTE COMPONENTS OF LOCAL WIND IN LIFTING SURFACE CO-ORDINATES.

      NST=[illegible](I-1)+1
      CALL [illegible](NST,1,7,1,40,0,EL1,EL2,EL3,VA,EV1)

C     CALCULATE LOADS GENERATED BY LIFTING SURFACE IN LIFTING SURFACE
C     CO-ORDINATES.

      IF(EV1(1)) [illegible],[illegible],[illegible]
      CONTINUE
      [illegible]=[illegible](EV1(3)/EV1(1))
      IF(EV1(1).GT.0.0.AND.ABS([illegible]).LT.0.2) GO TO 200

C     SMALL ANGLE-OF-ATTACK ASSUMPTIONS HAVE BEEN FOUND UNSATIS-
C     FACTORY.  PROCEED WITH THE LARGE ANGLE ANALYSIS.

      U=EV1(1)
      V=EV1(2)
      W=EV1(3)
      P=EV1(4)
      Q=EV1(5)
      R=EV1(6)

      IF(LOOP-2) 110,100,110
  100 CONTINUE
      DO 105 J=1,6
  105 FB(J)=0.0
      GO TO 900
  110 CONTINUE
```

```
[illegible]
[illegible]
[illegible]
PITCH=[illegible]
V[illegible]
CONTINUE

COMPUTE COMPONENTS OF LOCAL WIND IN LIFTING SURFACE CO-ORDINATES.

KST=6*(I-1)+1
CALL EULER(KST,1,2,1,40,6,EL1,EL2,EL3,VA,RV1)

CALCULATE LOADS GENERATED BY LIFTING SURFACE IN LIFTING SURFACE
CO-ORDINATES.

IF(RV1(1)) 105,107,105
CONTINUE
ABSWU=ABS(RV1(3)/RV1(1))
IF(RV1(1).GT.0.0.AND.ABSWU.LT.0.2) GO TO 200

SMALL ANGLE-OF-ATTACK ASSUMPTIONS HAVE BEEN FOUND UNSATIS-
FACTORY.  PROCEED WITH THE LARGE ANGLE ANALYSIS.

U=RV1(1)
V=RV1(2)
W=RV1(3)
P=RV1(4)
Q=RV1(5)
R=RV1(6)

IF(U**2+W**2) 110,108,110
CONTINUE
DO 109 J=1,6
RV2(J)=0.0
GO TO 999
CONTINUE

COMPUTE THE AIRSPEED (BIGV), AND THE ANGLE OF ATTACK (ALFA).
ALFA IS THE ANGLE BETWEEN THE LIFTING SURFACE LINE OF ZERO
LIFT AND THE RELATIVE WIND VECTOR.  ALFA LIES BETWEEN PLUS
AND MINUS PI.

IF(U)130,120,130
ALFA=SIGN(1.5708,W)
BIGV2=W**2
GO TO 140
BIGV2=U**2+W**2
BIGV=BIGV2**0.5
AAS=ABS(W/BIGV)
IF(AAS.GE.1.0) ALFA=SIGN(1.5708,W)
```

```
IF(AAG.GE.1.0) GO TO 132
ALFA=ASIN(W/BIGV)
CONTINUE
IF(U.LT.0.0.AND.ALFA.GE.0.0) ALFA=3.1416-ALFA
IF(U.LT.0.0.AND.ALFA.LT.0.0) ALFA=-3.1416-ALFA
CONTINUE

COMPUTE THE DRAG COEFFICIENT.

ABSALF=ABS(ALFA+ANGLE)
CDSTL=1.5-.812*(1.57-ABSALF)**2
IF(ABSALF.GE.0.30.AND.ABSALF.LE.2.80) CD=CDSTL
IF(ABSALF.GT.0.30.AND.ABSALF.LE.2.80) GO TO 150
IF(ABSALF.GT.2.80) GO TO 140
CD=CD0+CD2*ABSALF**2
IF(CDSTL.GT.CD) CD=CDSTL
GO TO 150
CONTINUE
CD=2.75*(CD0+CD2*(ABSALF-3.1416)**2)
IF(CDSTL.GT.CD) CD=CDSTL
CONTINUE

COMPUTE THE LIFT COEFFICIENT.

ABSALF=ABS(ALFA)
ADOWN=0.0
CLMAX=1.10
IF(ABSALF.GE.0.30.AND.ABSALF.LE.2.80) GO TO 160
IF(ALFA.GT.2.80) ADOWN=3.1416
IF(ALFA.LT.-2.80) ADOWN=-3.1416
ALFAM=ALFA-ADOWN
CLEXT=SIGN(CLMAX,ALFAM)
CL=A*ALFAM
ACL=ABS(CL)
IF(ACL.GT.CLMAX) CL=CLEXT
GO TO 170
CONTINUE
CL=CLMAX*SIN(2.0*ALFA)
CONTINUE

CALCULATE THE CHORD AND NORMAL-TO-CHORD FORCES.

SA=SIN(ALFA)
CA=COS(ALFA)
QDYN=0.5*RHO*BIGV2
XU=QDYN*(CL*SA-CD*CA)*SW
```

```
      CD=CD1*(1.0-.0110*(1.57-ABSALF)**2
      IF(ABSALF.GT.0.35.AND.ABSALF.LE.2.80) CD=CDSTL
      IF(ABSALF.GT.0.35.AND.ABSALF.LE.2.80) GO TO 150
      IF(ABSALF.GT.2.80) GO TO 130
      CD=CD0+CD2*ABSALF**2
      IF(CDSTL.GT.CD) CD=CDSTL
      GO TO 150
  130 CONTINUE
      CD=2.75*(CD0+CD2*(ABSALF-3.1416)**2)
      IF(CDSTL.GT.CD) CD=CDSTL
  150 CONTINUE
C
C     COMPUTE THE LIFT COEFFICIENT.
C
      ABSALF=ABS(ALFA)
      ADD=0.0
      CLMAX=1.10
      IF(ABSALF.GT.0.35.AND.ABSALF.LE.2.80) GO TO 160
      IF(ALFA.GT.2.80) ADD=3.1416
      IF(ALFA.LT.-2.80) ADD=-3.1416
      ALFAM=ALFA-ADD
      CLLIM=SIGN(CLMAX,ALFAM)
      CL=CLA*ALFAM
      ACL=ABS(CL)
      IF(ACL.GT.CLMAX) CL=CLLIM
      GO TO 170
  160 CONTINUE
      CL=CLMAX*SIN(2.0*ALFA)
  170 CONTINUE
C
C     CALCULATE THE CHORD AND NORMAL-TO-CHORD FORCES.
C
      SA=SIN(ALFA)
      CA=COS(ALFA)
      QLVR=0.5*S*RHO*BIGV2
      XL=QLVR*(CL*SA-CD*CA)*SU
      ZL=QLVR*(-CL*CA-CD*SA)*SU
      DO 185 J=1,6
  185 FV2(J)=0.0
      FV2(1)=XL
      FV2(3)=ZL
      GO TO 300
  190 CONTINUE
C
C     SMALL ANGLE APPROXIMATIONS ARE APPROPRIATE- PROCEED WITH
C     THE SMALL ANGLE ANALYSIS.
C
      CUW=RV1(3)-ALCLD*RV1(1)
      FV2(1)=F*(A*RV1(3)**2-CD0*RV1(1)**2-CD1*RV1(1)*CUW-CD2*CUW**2)
      FV2(2)=0.0
```

```
RV2(3)=[illegible]
RV2(4)=[illegible]
RV2(5)=[illegible]
RV2(6)=[illegible]
CONTINUE

ROTATE LIFTING SURFACE LOADS BACK TO AIRPLANE CO-ORDINATES.

CALL EULER(1,KST,2,2,6,6A,EL1,EL2,EL3,RV2,FD)
RETURN
END
```

```
      SUBROUTINE ROTOR(I,NOPT,V0,VI,VIDOT,C,FR)
C
C     IF NOPT=1, CALCULATE A NEW BLADE-MOTION MATRIX. IF NOPT=2, USE
C     THE OLD BLADE-MOTION MATRIX. NOPT MUST BE 1 THE FIRST TIME THRU
C     THIS ROUTINE FOR EACH ROTOR ON THE VEHICLE.
C
C     IN THE ARRAYS BLTMP AND BLPROP, THE FIRST SUBSCRIPT REPRESENTS THE
C     BLADE POINT NUMBER. THE SECOND SUBSCRIPT INDICATES THE PHYSICAL
C     QUANTITY REPRESENTED BY THE PARTICULAR ARRAY ELEMENT. 1=MASS,
C     2=RADIAL POSITION(S), 3=INITIAL MODESHAPE COORDINATE, 4=DYNAMIC
C     MODESHAPE COORDINATE. BLPROP(I,5) IS THE BLADE CHORD AT
C     STATION I.
C
      DIMENSION V0(40),VI(40),VIDOT(40),C(37),FR(40),FM(6),A(3)
C
      COMMON/[illegible]/ [illegible],CAPTO,CAPTA,CAPTU,DELT0,DELT1,DELT2,DELT3,
     1 [illegible]
     2 [illegible]
     3 [illegible],SPACE(8),
     4 [illegible],SPACE7(91),
     5 [illegible]
      COMMON/[illegible]/[illegible](6),[illegible](6),[illegible](6)
      COMMON/[illegible]/[illegible](8,2),[illegible](8),[illegible](8),N,NPASS, NDIRCT,NEX,NITER
      COMMON/[illegible]/[illegible],[illegible],TAS,[illegible],PSIDOT,[illegible],ROLLRT,NOOT,LC,
     1 [illegible],IT(4),[illegible](3,3)
      COMMON/[illegible]/[illegible]
      COMMON/[illegible]/[illegible],BETA(50,8),BETADT(50,8)
C
      NTAU(I)=NTAU(I)+1
      IF(NTAU(I).GT.1) GO TO 50
C
C     COMPUTE THE CONSTANT MASS INTEGRALS USED IN THE ROTOR EQUATIONS.
C
      [illegible]=0.0
      [illegible]=0.0
      [illegible]=0.0
      [illegible]=0.0
      [illegible]=0.0
      DO 10 J=2,[illegible]
      [illegible](J,1)
      [illegible](J-1,1)
      [illegible](J,2)
      [illegible](J-1,2)
      [illegible](J,4)
      [illegible](J-1,4)
```

```
2-RADIAL POSITION(S), 3-INITIAL MODESHAPE COORDINATE, 4-DYNAMIC
MODESHAPE COORDINATE.  BLDPRP(I,5) IS THE BLADE CHORD AT
STATION I.

DIMENSION VA(40),VI(40),VIDOT(40),C(17),FR(40),FM(6),A(3)

COMMON/PHYSCS/ OMEG,CAPRO,CAPRA,CAPFO,DELT0,DELT1,DELT2,DELT3,
1 THET1,R,RR,SA,E1,E2,E3,DELTE,VSS,DPSI,PR,PRO,
2 MASS,[illegible],OM11,OM12,OMS,SMS2,OM1,[illegible],SJM,COT,SGT,SOMG,
3 COPSI,SOPSI,[illegible],NC,[illegible],PHI1(2,2),SPACE(8),
4 BLDPRP(20,5),PNTMS(10,5),SPACE2(51),
5 [illegible],SPCE1(7),[illegible],NPNTS(8)
[illegible]/EV1(6),EV2(6),EV3(6)
COMMON/[illegible]/[illegible](4,2),NTYPE(8),NTHRU(8),N,NPASS, NDIRCT,NEX,NITER
COMMON/[illegible]/UT,XCG,YCG,ZCG,TAS,[illegible],PSIDCT,PTCHDT,ROLLDT,HDOT,LC,
1 [illegible],IT(4),GIPRTA(3,3)
COMMON/[illegible]/[illegible]
COMMON/BLDHST/ICNIST,BETA(50,8),BETADT(50,8)

NTHRU(1)=NTHRU(1)+1
IF(NTHRU(1).GT.1) GO TO 50

COMPUTE THE CONSTANT MASS INTEGRALS USED IN THE ROTOR EQUATIONS.

OMS=0.0
OM11=0.0
OM12=0.0
OM1=0.0
SJM=0.0
DO 10 J=2,NBS
SMJ=BLDPRP(J,1)
SMJM1=BLDPRP(J-1,1)
SJ=BLDPRP(J,2)
SJM1=BLDPRP(J-1,2)
Z1J=BLDPRP(J,4)
Z1JM1=BLDPRP(J-1,4)
DS=SJ-SJM1
OMS=OMS+0.5*(SMJ*Z1J**2+SMJM1*Z1JM1**2)*DS
OM11=OM11+0.5*(SMJ*Z1J+SMJM1*Z1JM1)*DS
OM12=OM12+0.5*(SMJ*Z1J*SJ+SMJM1*Z1JM1*SJM1)*DS
OM1=OM1+0.5*(SMJ*SJ+SMJM1*SJM1)*DS
SJM=SJM+0.5*(SMJ*SJ**2+SMJM1*SJM1**2)*DS

ADD POINT MASS CONTRIBUTIONS TO INTEGRALS.

DO 15 J=1,NPM
DMASS=PNTMS(J,1)
S=PNTMS(J,2)
Z1=PNTMS(J,4)
OMS=OMS+DMASS*Z1**2
```

```
[illegible]
[illegible]
[illegible]
[illegible]

IF THIS IS A RIGID-BLADED ROTOR, SET SMS=1.0, SMI1=0.0 AND
SMI2=0.0 TO AVOID ERROR CALCULATIONS IN SWEEP.

IF(NTYPE(I).NE.4) GO TO 20
SMS=1.0
SMI1=0.0
SMI2=0.0
CONTINUE

CALCULATE GENERAL CONSTANTS FOR ROTOR EQUATIONS.

[illegible]
DPSI=(3.14159*2.0)/BNAB
SDPSI=SIN(DPSI)
CDPSI=COS(DPSI)
[illegible]
SDT=SIN(PHLDG*DPSI)
CDT=COS(PHLDG*DPSI)
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
FORMAT(1X,8F15.8)
FORMAT(1X,12E10.4)
CONTINUE
[illegible]
[illegible]
[illegible]

ROTATE THE APPROPRIATE SUB-VECTORS OF VA,VI AND VIDOT TO ALLIGN
WITH ROTOR SYSTEM COORDINATES.

KST=6*(I-1)+1
CALL EULER(KST,1,2,1,48,6,E1,E2,E3,VA,RV1)
CALL EULER(KST,1,2,1,48,6,E1,E2,E3,VI,RV2)
CALL EULER(KST,1,2,1,48,6,E1,E2,E3,VIDOT,RV3)

DETERMINE THE ROTOR CONTROL SETTINGS.

[illegible]
```

```
CONTINUE

CALCULATE GENERAL CONSTANTS FOR ROTOR EQUATIONS.

SNASONAS
DPSI=(3.14159*2.0)/SNAS
SDPSI=SIN(DPSI)
CDPSI=COS(DPSI)
[illegible]/2.0
SNT=SIN(PHASE*DPSI)
CNT=COS(PHASE*DPSI)
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
FORMAT(1X,6E15.8)
FORMAT(1X,12E10.4)
CONTINUE
[illegible]
[illegible]
[illegible]

ROTATE THE APPROPRIATE SUB-VECTORS OF VA,VI AND VIDOT TO ALLIGN
WITH ROTOR SYSTEM COORDINATES.

KST=60*(I-1)+1
CALL ELLEM(KST,1,2,1,40,6,E1,E2,E3,VA,RV1)
CALL ELLEM(KST,1,2,1,40,6,E1,E2,E3,VI,RV2)
CALL ELLEM(KST,1,2,1,40,6,E1,E2,E3,VIDOT,RV3)

DETERMINE THE ROTOR CONTROL SETTINGS.

INDC=NC(I,1)
A(1)=C(INDC)
A(2)=C(INDC+1)
A(3)=C(INDC+2)

IF THIS IS A RIGID-BLADED ROTOR, PROCEED TO THE FORCE COMPUTATION.

IF(NTYPE(I).EQ.4) GO TO 575

MAKE THE FIRST PASS THROUGH THE BLADE MOTION EQUATIONS.

[illegible]
[illegible]
CALL S[illegible](1,RV1,RV2,RV3,A,FM,B,BD)
[illegible]
```

```
[illegible]

IF A NEW BLADE MOTION MATRIX IS NOT TO BE GENERATED, GO TO THE
COMPUTATION WHICH UPDATES THE INITIAL CONDITIONS.  OTHERWISE,
FIND ZZR.

IF(NOPT.EQ.2) GO TO 450
B=B0+PB
[illegible]
CALL SWEEP(1,RV1,RV2,RV3,A,FM,B,BD)
ZZ11= (B-B2PI)/PB
ZZ21= (BD-BD2PI)/PB
B=B0
[illegible]
CALL SWEEP(1,RV1,RV2,RV3,A,FM,B,BD)
ZZ12= (B-B2PI)/PBD
ZZ22= (BD-BD2PI)/PBD

FORM THE INVERSE OF THE ARRAY (ZZR-I) AND NAME THE RESULT ZMI1.

ZZ11=ZZ11-1.0
ZZ22=ZZ22-1.0
DETER=ZZ11*ZZ22-ZZ12*ZZ21
ZMI1(1,1)=ZZ22/DETER
ZMI1(1,2)=-ZZ12/DETER
ZMI1(2,1)=-ZZ21/DETER
ZMI1(2,2)=ZZ11/DETER
CONTINUE

CALCULATE THE UPDATED INITIAL CONDITIONS.

ZK1=B2PI-B0
ZK2=BD2PI-BD0
B0=B0-(ZMI1(1,1)*ZK1+ZMI1(1,2)*ZK2)
BD0=BD0-(ZMI1(2,1)*ZK1+ZMI1(2,2)*ZK2)
IF(NOPT.EQ.2) GO TO 475
B=B0
BD=BD0
CONTINUE
CONTINUE

DETERMINE INDXSP (OPTION ON FLEXIBLE BLADE), AND COMPUTE ROTOR
LOADS

[illegible]
[illegible]
```

```
ZZ011= (B-R2PI)/PB
ZZ021= (BO-RD2PI)/PB
B=PB
BD=FDB+PBD
CALL SWEEP(1,RV1,RV2,RV3,A,FH,B,BD)
ZZ012= (B-R2PI)/PBD
ZZ022= (BD-RD2PI)/PBD

FORM THE INVERSE OF THE ARRAY (ZZ0-I) AND NAME THE RESULT ZM1I.

ZZ011=ZZ011-1.0
ZZ022=ZZ022-1.0
DETER=ZZ011*ZZ022-ZZ012*ZZ021
ZM1I(1,1)=ZZ022/DETER
ZM1I(1,2)=-ZZ012/DETER
ZM1I(2,1)=-ZZ021/DETER
ZM1I(2,2)=ZZ011/DETER
CONTINUE

CALCULATE THE UPDATED INITIAL CONDITIONS.

ZK1=R2PI-B
ZK2=RD2PI-BD
PU=PB-(ZM1I(1,1)*ZK1+ZM1I(1,2)*ZK2)
PBDU=PBD-(ZM1I(2,1)*ZK1+ZM1I(2,2)*ZK2)
IF(NOPT.EQ.2) GO TO 475
BO=PU
BDO=PBDU
CONTINUE
CONTINUE

DETERMINE INDXSP (OPTION ON FLEXIBLE BLADE), AND COMPUTE ROTOR
LOADS

INDXSP=2
IF(NTYPE(I).EQ.4)INDXSP=3
B=PU
BD=PBDU
CALL SWEEP(INDXSP,RV1,RV2,RV3,A,FH,B,BD)

ROTATE THE ROTOR LOADS BACK TO VEHICLE COORDINATES.

CALL EULER(1,KST,2,2,4,48,E1,E2,E3,FH,FR)
RETURN
END
```

```
      SUBROUTINE VELCTY(T,VNOT,S,NDIRCT,SPHI,CPHI,STH,CTH)

      THE INDEX NOPTRM INDICATES WHICH VARIABLES ARE REPRESENTED BY THE
      GIVEN QUANTITIES PTCHRT AND ROLLRT

         NOPTRM          QUANTITIES REPRESENTED BY PTCHRT AND ROLLRT

           1                                THETAD, PHID
           2                                THETAD, P
           3                                Q,PHID
           4                                Q, P

      NDIRCT=0 FOR FORWARD FLIGHT AND 1 FOR BACKWARD FLIGHT.

      DIMENSION T(6),VNOT(9),SFE(3),S(6)

      COMMON/SPEC/WT,XCG,YCG,ZCG,TAS,RHO,PSIDOT,PTCHRT,ROLLRT,HDOT,LC,
     1 NOPTRM,IT(6),QINRTA(3,3)

      DETERMINE THETA, PHI AND V.

      IF(NDIRCT.EQ.0) QDIRCT=1.0
      IF(NDIRCT.EQ.1)QDIRCT=-1.0
      JNOT=LC-3
      DO 10 I=1,3
      NEL=LC+4-I
      DO 5 J=1,6
      IF(IT(7-J).NE.NEL) GO TO 5
      SFE(I)=T(7-J)
      GO TO 10
      CONTINUE
      SFE(I)=VNOT(JNOT)
      JNOT=JNOT-1
      CONTINUE
      V=SFE(1)
      PHI=SFE(2)
      THETA=SFE(3)

      TEST TO SEE IF THETA IS WITHIN PLUS OR MINUS 90 DEGREES.
```

```
      ATH=ABS(THETA)
      IF(ATH-1.5700)225,220,220
      CTV=SIGN(1.5700,THETA)
      THETA=CTV
      DO 223 J=1,6
      IF(IT(J).NE.LC+1) GO TO 223
      T(J)=THETA
      CONTINUE
      WRITE(6,224)
      FORMAT(/,1X,'VELCTY HAD TO CHANGE THE VALUE OF THETA BECAUSE ',
     1 'IT WAS OUT OF THE PLUS OR MINUS 90 DEGREE LIMIT',/)
      CONTINUE

      TEST TO SEE IF PHI IS WITHIN PLUS OR MINUS 90 DEGREES.

      APH=ABS(PHI)
      IF(APH-1.57)235,230,230
      CTV=SIGN(1.57,PHI)
      PHI=CTV
      DO 233 J=1,6
      IF(IT(J).NE.LC+2) GO TO 233
      NEL=LC+4-I
      DO 5 J=1,6
      IF(IT(7-J).NE.NEL) GO TO 5
      SEE(I)=T(7-J)
      GO TO 10
      CONTINUE
      SEE(I)=VNOT(JNOT)
      JNOT=JNOT-1
      CONTINUE
      V=SEE(1)
      PHI=SEE(2)
      THETA=SEE(3)

      TEST TO SEE IF THETA IS WITHIN PLUS OR MINUS 90 DEGREES.

      ATH=ABS(THETA)
      IF(ATH-1.5700)225,220,220
      CTV=SIGN(1.5700,THETA)
      THETA=CTV
      DO 223 J=1,6
      IF(IT(J).NE.LC+1) GO TO 223
      T(J)=THETA
      CONTINUE
      WRITE(6,224)
      FORMAT(/,1X,'VELCTY HAD TO CHANGE THE VALUE OF THETA BECAUSE ',
     1 'IT WAS OUT OF THE PLUS OR MINUS 90 DEGREE LIMIT',/)
      CONTINUE
```

```
TEST TO SEE IF PHI IS WITHIN PLUS OR MINUS 90 DEGREES.

APH=ABS(PHI)
IF(APH-1.57)235,230,230
CTV=SIGN(1.57,PHI)
PHI=CTV
DO 233 J=1,6
IF(IT(J).NE.LC+2) GO TO 233
T(J)=PHI
CONTINUE
WRITE(6,234)
FORMAT(/,1X,'VELCTY HAD TO CHANGE THE VALUE OF PHI BECAUSE ',
1 'IT WAS OUT OF THE PLUS OR MINUS 90 DEGREE LIMIT',/)
CONTINUE

CALCULATE SINES AND COSINES OF THETA AND PHI.

SPHI=SIN(PHI)
CPHI=COS(PHI)
STH=SIN(THETA)
CTH=COS(THETA)


SET U,V AND W TO ZERO IF THE SPEED IS ZERO (CASE 1).

THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
IF(TAS.NE.0.0) GO TO 245
U=0.0
V=0.0
W=0.0
GO TO 450
CONTINUE

CALCULATE U,V AND W DIRECTLY AS FUNCTIONS OF HDOT, PHI AND THETA,
IF THE SPEED=HDOT (CASE 2).

AHD=ABS(HDOT)
IF(AHD-TAS)250,247,250
U=STH*HDOT
V=-SPHI*CTH*HDOT
W=-CPHI*CTH*HDOT
GO TO 450
CONTINUE

COMPUTE C AND R-SQUARED.

RDCL=SQRT(TAS**2-HDOT**2)
C=(V*SPHI*CTH*HDOT)/RDCL
RSQD=CPHI**2+(SPHI*STH)**2
```

```
DETERMINE WHETHER THIS IS CASE 3 OR 4 (DEPENDING ON THE
RELATIVE SIZES OF C-SQUARED AND R-SQUARED).

IF(C**2.GT.RSQD) GO TO 275

COMPUTE CCY AND SCY, LEAVING V UNALTERED (CASE 3).

RDCL2=WDIRCT*SQRT(RSQD-C**2)
CCY=(RDCL2*CPHI+C*SPHI*STH)/RSQD
SCY=(-C*CPHI+RDCL2*SPHI*STH)/RSQD
GO TO 290

COMPUTE CCY AND SCY WITH THE REQUIREMENT TO CHANGE THE GIVEN VALUE
OF V (CASE 4).

CONTINUE
CAPR=SQRT(RSQD)
C=[illegible]
[illegible]

[illegible] AS [illegible] PHI AND THETA,
[illegible] (CASE 2).

[illegible]
IF([illegible]-TAS)25[illegible],247,250
[illegible]
[illegible]
[illegible]HDOT
GO TO 450
CONTINUE

COMPUTE C AND R-SQUARED.

RDCL=SQRT(TAS**2-HDOT**2)
C=(V+SPHI*CTH*HDOT)/RDCL
RSQD=CPHI**2+(SPHI*STH)**2

DETERMINE WHETHER THIS IS CASE 3 OR 4 (DEPENDING ON THE
RELATIVE SIZES OF C-SQUARED AND R-SQUARED).

IF(C**2.GT.RSQD) GO TO 275

COMPUTE CCY AND SCY, LEAVING V UNALTERED (CASE 3).

RDCL2=WDIRCT*SQRT(RSQD-C**2)
CCY=(RDCL2*CPHI+C*SPHI*STH)/RSQD
SCY=(-C*CPHI+RDCL2*SPHI*STH)/RSQD
GO TO 290

COMPUTE CCY AND SCY WITH THE REQUIREMENT TO CHANGE THE GIVEN VALUE
OF V (CASE 4).
```

```
CONTINUE
CAPR=SQRT(RSQD)
C=QDIRCT*CAPR
CCY=(C*SPHI*STH)/RSQD
SCY=-(C*CPHI)/RSQD

COMPUTE U,V AND W FOR CASES 3 OR 4.

CONTINUE
U=(CTH*CCY)*RDCL+(STH)*HDOT
V=(SPHI*STH*CCY-CPHI*SCY)*RDCL-(SPHI*CTH)*HDOT
W=(SPHI*SCY+CPHI*STH*CCY)*RDCL-(CPHI*CTH)*HDOT
CONTINUE

DETERMINE P Q AND R.

IF(NOPTRM.EQ.3.OR.NOPTRM.EQ.4)Q=PTCHRT
IF(NOPTRM.EQ.2.OR.NOPTRM.EQ.4)P=ROLLRT

IF(NOPTRM.EQ.1.OR.NOPTRM.EQ.3)P=ROLLRT-PSIDOT*STH
IF(NOPTRM.EQ.1.OR.NOPTRM.EQ.2)Q=PTCHRT*CPHI+PSIDOT*CTH*SPHI
R=(PSIDOT*CTH-Q*SPHI)/CPHI
S(1)=U
S(2)=V
S(3)=W
S(4)=P
S(5)=Q
S(6)=R
RETURN
END
```

```
      SUBROUTINE CONTRL(T,VNOT,C)

      DIMENSION T(6),VNOT(9),C(12)

      COMMON/SPEC/WT,XCG,YCG,ZCG,TAS,RHO,PSIDOT,PTCHRT,ROLLRT,HDOT,LC,
     1 NOPTRM,IT(6),QINRTA(3,3)

      FILL OUT THE CONTROL COLUMN.

      JNOT=1
      DO 15 I=1,LC
      DO 4 J=1,6
      IF(IT(J).NE.I) GO TO 4
      C(I)=T(J)
      GO TO 14
      CONTINUE
      C(I)=VNOT(JNOT)
      JNOT=JNOT+1
      CONTINUE
      CONTINUE
      RETURN
      END
```

```
      SUBROUTINE WASH(VA,VI,VIDOT,F,W,X,IX,JX,A,NEX)
C
      DIMENSION VA(48),VI(48),VIDOT(48),F(48),W(48)
      DIMENSION X(500),IX(500),JX(500),A(6,6,8),D(48)
C
      COMMON/INDECS/NC(8,2),NTYPE(8),NTHRU(8),N,NPASS, NDIRCT,NEX,NITER
      COMMON/SPEC/WT,XCG,YCG,ZCG,TAS,RHO,PSIDOT,PTCHRT,ROLLRT,HDOT,LC,
     1 NOPTRM,IT(6),QINRTA(3,3)
C
      NELW=6*N
      DO 2 K=1,NELW
    2 W(K)=0.0
      FACT=1.0/(2.0*RHO)
      DO 10 I=1,N
      INDE=6*(I-1)
      ARG=VA(INDE+1)**2+VA(INDE+2)**2+VA(INDE+3)**2
      VAT=SQRT(ARG)
      DO 5 J=1,6
      NJ=INDE+J
      D(NJ)=0.0
      THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
      IF(VAT.EQ.0.0) GO TO 5
      DO 4 K=1,6
      NK=INDE+K
      D(NJ)=D(NJ)-(A(J,K,I)*F(NK)*FACT)/VAT
    4 CONTINUE
    5 CONTINUE
      DO 20 LEX=1,NEX
      I=IX(LEX)
      J=JX(LEX)
      FACTOR=X(LEX)
   20 W(I)=W(I)+FACTOR*D(J)
      RETURN
      END
```

```
      SUBROUTINE FCERQD(S,SPHI,CPHI,STH,CTH,FRQD)
      DIMENSION S(6),FRQD(6)
      COMMON/SPEC/WT,XCG,YCG,ZCG,TAS,RHO,PSIDOT,PTCHRT,ROLLRT,HDOT,LC,
     1 NOPTRM,IT(6),QINRTA(3,3)
      QMASS=WT/32.2
      UCG=S(1)+(S(5)*ZCG-S(6)*YCG)
      VCG=S(2)+(S(6)*XCG-S(4)*ZCG)
      WCG=S(3)+(S(4)*YCG-S(5)*XCG)
      PCG=S(4)
      QCG=S(5)
      RCG=S(6)
      XCGRQD=+WT*STH       +QMASS*(QCG*WCG-RCG*VCG)
      YCGRQD=-WT*CTH*SPHI+QMASS*(RCG*UCG-PCG*WCG)
      ZCGRQD=-WT*CTH*CPHI+QMASS*(PCG*VCG-QCG*UCG)
      T1=+QINRTA(1,1)*PCG-QINRTA(1,2)*QCG-QINRTA(1,3)*RCG
      T2=-QINRTA(2,1)*PCG+QINRTA(2,2)*QCG-QINRTA(2,3)*RCG
      T3=-QINRTA(3,1)*PCG-QINRTA(3,2)*QCG+QINRTA(3,3)*RCG
      QLCGR=T3*QCG-T2*RCG
      QMCGR=T1*RCG-T3*PCG
      QNCGR=T2*PCG-T1*QCG
      FRQD(1)=XCGRQD
      FRQD(2)=YCGRQD
      FRQD(3)=ZCGRQD
      FRQD(4)=QLCGR+(YCG*ZCGRQD-ZCG*YCGRQD)
      FRQD(5)=QMCGR+(ZCG*XCGRQD-XCG*ZCGRQD)
      FRQD(6)=QNCGR+(XCG*YCGRQD-YCG*XCGRQD)
      RETURN
      END
```

```
SUBROUTINE FORCF(K,NOPT,VA,VI,VIDOT,C,FR,PK,INTS)

IF NOPT=1,RETURN ALL NEW ELEMENTS OF FR. IF NOPT=2, RETURN ONLY
ELEMENTS FR(6K-5) TO FR(6K). K IS THE ELEMENT SEQUENCE NUMBER.

DIMENSION VA(48),VI(48),VIDOT(48),C(12),FR(48)
DIMENSION PK(250,8),INTS(10,8)

COMMON/PHYSCS/P(250),INTG(10),NPK(8),NINTS(8)
COMMON/INDECS/NC(8,2),NTYPE(8),NTHRU(8),N,NPASS, NDIRCT,NEX,NITER
COMMON/BLDHST/IOHIST,BETA(50,8),BETADT(50,8)
COMMON/DEBUG/NDXS

IF(NOPT.EQ.2)GO TO 550
DO 475 I=1,N
NFARP=NPK(I)
NFARI=NINTS(I)

DO 26 IFAR=1,NFARP
P(IFAR)=PK(IFAR,I)
IF(NFARI)27,29,27
CONTINUE
DO 28 JFAR=1,NFARI
INTG(JFAR)=INTS(JFAR,I)
CONTINUE
IF(NTYPE(I)-2)25,50,75
CALL LIFT(I,VA,C,FR)
GO TO 450
CALL BODY(I,VA,FR)
GO TO 450
CALL RCTOR(I,NOPT,VA,VI,VIDOT,C,FR)
CONTINUE
DO 456 IFAR=1,NFARP
PK(IFAR,I)=P(IFAR)
IF(NFARI)457,475,457
CONTINUE
DO 458 JFAR=1,NFARI
INTS(JFAR,I)=INTG(JFAR)
CONTINUE
GO TO 1200
CONTINUE
NFARP=NPK(K)
NFARI=NINTS(K)
DO 626 IFAR=1,NFARP
P(IFAR)=PK(IFAR,K)
IF(NFARI)627,629,627
CONTINUE
```

```
      DO 628 JFAR=1,NFARI
      INTG(JFAR)=INTS(JFAR,K)
      CONTINUE
      IF(NTYPE(K)-2)625,650,675
      CALL LIFT(K,VA,C,FR)
      GO TO 1000
      CALL BODY(K,VA,FR)
      GO TO 1000
      CALL ROTOR(K,NOPT,VA,VI,VIDOT,C,FR)
      CONTINUE
      DO 756 IFAR=1,NFARP
      PK(IFAR,K)=P(IFAR)
      IF(NFARI)757,1200,757
      CONTINUE
      DO 758 JFAR=1,NFARI
      INTS(JFAR,K)=INTG(JFAR)
      CONTINUE
      CALL BODY(I,VA,FR)
      GO TO 450
      CALL ROTOR(I,NOPT,VA,VI,VIDOT,C,FR)
      CONTINUE
      DO 456 IFAR=1,NFARP
      PK(IFAR,I)=P(IFAR)
      IF(NFARI)457,475,457
      CONTINUE
      DO 458 JFAR=1,NFARI
      INTS(JFAR,I)=INTG(JFAR)
      CONTINUE
      GO TO 1200
      CONTINUE
      NFARP=NPK(K)
      NFARI=NINTS(K)
      DO 626 IFAR=1,NFARP
      P(IFAR)=PK(IFAR,K)
      IF(NFARI)627,629,627
      CONTINUE
      DO 628 JFAR=1,NFARI
      INTG(JFAR)=INTS(JFAR,K)
      CONTINUE
      IF(NTYPE(K)-2)625,650,675
      CALL LIFT(K,VA,C,FR)
      GO TO 1000
      CALL BODY(K,VA,FR)
      GO TO 1000
      CALL ROTOR(K,NOPT,VA,VI,VIDOT,C,FR)
      CONTINUE
      DO 756 IFAR=1,NFARP
      PK(IFAR,K)=P(IFAR)
      IF(NFARI)757,1200,757
      CONTINUE
      DO 758 JFAR=1,NFARI
      INTS(JFAR,K)=INTG(JFAR)
      CONTINUE
      RETURN
      END
```

```
SUBROUTINE MTXADD(A,B,C,NR,NC,NRDIMA,NRDIMB,NRDIMC,NCODE)

THIS MATRIX ADDITION SUBROUTINE ADDS OR SUBTRACTS B TO OR FROM A
TO YIELD C. PROCESS IS ADDITION IF NCODE IS 1. SUBTRACTION A-B=C
OCCURS FOR NCODE=2.

DIMENSION A(NRDIMA,NC),B(NRDIMB,NC),C(NRDIMC,NC)
IF(NCODE.EQ.1)GO TO 10
DO 5 I=1,NR
DO 4 J=1,NC
C(I,J)=A(I,J)-B(I,J)
CONTINUE
GO TO 20
CONTINUE
DO 15 I=1,NR
DO 14 J=1,NC
C(I,J)=A(I,J)+B(I,J)
CONTINUE
CONTINUE
RETURN
END
```

```
SUBROUTINE MTXMPY( A , B , C , NRA , NCA , NCB , NRDIMA , NRDIMB
1 , NRDIMC )
*******************************************************************
MATRIX MULTIPLICATION

  A( NRA , NCA ) * B( NCA , NCB ) =  C( NRA , NCB )
*******************************************************************
THIS MATRIX MULTIPLY SUBROUTINE IS A GENERAL ROUTINE  AND
  COMPUTES THE VECTOR INNER-PRODUCT ACCUMULATIONS IN
  DOUBLE PRECISION.
*******************************************************************
REAL  A( NRDIMA , NCA ) , B( NRDIMB , NCB ) , C( NRDIMC , NCB )
DOUBLE PRECISION TEMP
*******************************************************************
DO 10 I = 1, NRA
DO 10 J = 1, NCB
TEMP = 0.0
DO  5 K = 1, NCA
TEMP = TEMP + A(I,K) * B(K, J)
C(I, J) = TEMP
CONTINUE
RETURN
END
```

```
      SUBROUTINE MATINV(A,NA,N,DET,IRANK)
C
C     THIS SUBROUTINE INVERTS THE N BY N MATRIX A AND STORES THIS INVERSE
C     IN THE SAME STORAGE LOCATION ORIGINALLY OCCUPIED BY THE ORIGINAL
C     MATRIX A.    NA IS THE  DIMENSION FOR A SPECIFIED IN THE MAIN
C     PROGRAM.    THE PARAMETERS N, NA, AND THE MATRIX A MUST BE
C     SPECIFIED BEFORE THE SUBROUTINE IS CALLED.  SUBROUTINE MATINV
C     RETURNS THE DETERMINANT OF THE MATRIX (DET), AND THE RANK OF THE
C     MATRIX IRANK..
C
      DIMENSION A(2500),B(50),C(50),IROW(50),ICOL(50)
      DET=1.
      IRANK=0
C
C     LOCATE PIVOTAL ELEMENT
C
      DO 1500 K=1,N
      AMAX=0.
      KM1=K-1
      NM1=N-1
      DO 1050 J=KM1,NM1
      IJ=J*NA+K
      DO 1050 I=K,N
      ARG=A(IJ)*A(IJ)
      IF(ARG-AMAX) 1050,1050,1040
      AMAX=ARG
      IROW(K)=I
      ICOL(K)=J+1
      IJ=IJ+1
      IF(AMAX-1.E-20) 1060,1060,1100
      DET=0.
      IROW(K)=K
      ICOL(K)=K
      GO TO 1500
C
C     MOVE MAXIMUM ELEMENT TO PIVOTAL POSITION
C
      IRANK=IRANK+1
      IF(IROW(K)-K) 1110,1200,1110
      DET=-DET
      KKJ=IROW(K)
      KJ=K
      DO 1150 J=1,N
      TEMP=A(KKJ)
      A(KKJ)=A(KJ)
      A(KJ)=TEMP
      KKJ=KKJ+NA
```

```
      KJ=KJ+NA
      IF(ICOL(K)-K) 1210,1300,1210
      DET=-DET
      IKK=(ICOL(K)-1)*NA
      IK=(K-1)*NA
      DO 1250 I=1,N
      IKK=IKK+1
      IK=IK+1
      TEMP=A(IKK)
      A(IKK)=A(IK)
      A(IK)=TEMP

 EDUCE PIVOTAL ROW AND COLUMN

      KK=(K-1)*NA+K
      TEMP=A(KK)
      DET=DET*TEMP
      DO 1400 J=1,N
      JK=(K-1)*NA+J
      KJ=(J-1)*NA+K
      IF(J-K) 1350,1310,1350
      B(J)=1./TEMP
      C(J)=1.
 OVE MAXIMUM ELEMENT TO PIVOTAL POSITION

      IRANK=IRANK+1
      IF(IROW(K)-K) 1110,1200,1110
      DET=-DET
      KKJ=IROW(K)
      KJ=K
      DO 1150 J=1,N
      TEMP=A(KKJ)
      A(KKJ)=A(KJ)
      A(KJ)=TEMP
      KKJ=KKJ+NA
      KJ=KJ+NA
      IF(ICOL(K)-K) 1210,1300,1210
      DET=-DET
      IKK=(ICOL(K)-1)*NA
      IK=(K-1)*NA
      DO 1250 I=1,N
      IKK=IKK+1
      IK=IK+1
      TEMP=A(IKK)
      A(IKK)=A(IK)
    ) A(IK)=TEMP

 EDUCE PIVOTAL ROW AND COLUMN
```

```
KK=(K-1)*NA+K
TEMP=A(KK)
DET=DET*TEMP
DO 1400 J=1,N
JK=(K-1)*NA+J
KJ=(J-1)*NA+K
IF(J-K) 1350,1310,1350
B(J)=1./TEMP
C(J)=1.
GO TO 1400
B(J)=-A(KJ)/TEMP
C(J)=A(JK)
A(JK)=0.
A(KJ)=0.
DO 1450 J=1,N
IJ=(J-1)*NA
DO 1450 I=1,N
IJ=IJ+1
A(IJ)=A(IJ)+B(J)*C(I)
CONTINUE

REARRANGE MATRIX IN ORIGINAL FORM

DO 1700 KCYCLE=1,N
K=N+1-KCYCLE
IF(IROW(K)-K) 1510,1600,1510
IKK=(IROW(K)-1)*NA
IK=(K-1)*NA
DO 1550 I=1,N
IKK=IKK+1
IK=IK+1
TEMP=A(IKK)
A(IKK)=A(IK)
A(IK)=TEMP
IF(ICOL(K)-K) 1610,1700,1610
KKJ=ICOL(K)
KJ=K
DO 1650 J=1,N
TEMP=A(KKJ)
A(KKJ)=A(KJ)
A(KJ)=TEMP
KKJ=KKJ+NA
KJ=KJ+NA
CONTINUE
RETURN
END
```

```
SUBROUTINE EULER(KSTC,KSTR,NVCTS,LOPT,NDOC,NDOR,PSI,THETA,PHI,C,R)

THIS SUBROUTINE ROTATES (NVCTS)- 3 ELEMENT COLUMN VECTORS STRUNG
IN SERIES WITHIN THE COLUMN VECTOR C, THROUGH THREE CONSECUTIVE
EULER ROTATIONS PSI,THETA,PHI (OR IN THE OPPOSITE SENSE AND ORDER
DEPENDING ON (LOPT)). THE NUMBER OF THE FIRST ELEMENT OF THE FIRST
COLUMN VECTOR TO BE ROTATED WITHIN C IS (KSTC). THE ROTATED
VECTORS ARE RETURNED AS A SERIES STRING WITHIN R.  (KSTR) IS DEFI-
NED ANALOGOUSLY TO (KSTC), EXCEPT (KSTR) RELATES TO R.  THE LOPT
OPTION INDEX DEFINES THE FOLLOWING OPTIONS.

    LOPT                    OPTION

     1      ROTATE THROUGH LARGE ANGLES PSI,THETA,PHI

     2      ROTATE THROUGH LARGE ANGLES -PHI,-THETA,-PSI.

     3      ROTATE THROUGH SMALL ANGLES PSI,THETA,PHI.

     4      ROTATE THROUGH SMALL ANGLES -PHI,-THETA,-PSI.

DIMENSION C(NDOC),R(NDOR),E(3,3)

COMPUTE SINES AND COSINES OF THE EULER ANGLES, OBSERVING OPTIONS.

IF(LOPT.EQ.1.OR.LOPT.EQ.2)GO TO 10
SCV=PSI

CCV=1.0-0.5*PSI**2
STH=THETA
CTH=1.0-0.5*THETA**2
SFI=PHI
CFI=1.0-0.5*PHI**2
GO TO 15
SCV=SIN(PSI)
CCV=COS(PSI)
STH=SIN(THETA)
CTH=COS(THETA)
SFI=SIN(PHI)
CFI=COS(PHI)
CONTINUE

ASSEMBLE THE EULER ROTATIONAL MATRIX.

E(1,1)=CTH*CCV
E(1,2)=CTH*SCV
E(1,3)=-STH
E(2,1)=SFI*STH*CCV-CFI*SCV
E(2,2)=CFI*CCV+STH*SCV*SFI
E(2,3)=SFI*CTH
E(3,1)=SFI*SCV+CFI*STH*CCV
E(3,2)=CFI*STH*SCV-SFI*CCV
E(3,3)=CFI*CTH
```

```
DETERMINE THE ORDER-AND-SENSE OPTION.

IF(LOPT.EQ.1.OR.LOPT.EQ.3)GO TO 25
W=E(1,2)
E(1,2)=E(2,1)
E(2,1)=W
W=E(1,3)
E(1,3)=E(3,1)
E(3,1)=W
W=E(2,3)
E(2,3)=E(3,2)
E(3,2)=W
CONTINUE

EULER MATRIX COMPLETE. NOW ROTATE THE APPROPRIATE VECTORS.

DO 100 M=1,NVCTS
DO 50 I=1,3
IM=3*(M-1)+I+KSTR-1
R(IM)=0.0
SFI=SIN(PHI)
CFI=COS(PHI)
CONTINUE

ASSEMBLE THE EULER ROTATIONAL MATRIX.

E(1,1)=CTH*CCV
E(1,2)=CTH*SCV
E(1,3)=-STH
E(2,1)=SFI*STH*CCV-CFI*SCV
E(2,2)=CFI*CCV+STH*SCV*SFI
E(2,3)=SFI*CTH
E(3,1)=SFI*SCV+CFI*STH*CCV
E(3,2)=CFI*STH*SCV-SFI*CCV
E(3,3)=CFI*CTH

DETERMINE THE ORDER-AND-SENSE OPTION.

IF(LOPT.EQ.1.OR.LOPT.EQ.3)GO TO 25
W=E(1,2)
E(1,2)=E(2,1)
E(2,1)=W
W=E(1,3)
E(1,3)=E(3,1)
E(3,1)=W
W=E(2,3)
E(2,3)=E(3,2)
E(3,2)=W
CONTINUE
```

```
EULER MATRIX COMPLETE. NOW ROTATE THE APPROPRIATE VECTORS.
DO 100 M=1,NVCTS
DO 50 I=1,3
IM=3*(M-1)+I+NSTR-1
A(IM)=0.0
DO 40 J=1,3
IMO=3*(M-1)+J+NSTC-1
A(IM)=A(IM)+E(I,J)*C(IMO)
CONTINUE
CONTINUE
RETURN
END
```

LITERATURE CITED

1. Duncan, W. J., Control and Stability of Aircraft, Cambridge University Press, 1952.

2. Seckel, Edward E., Stability and Control of Airplanes and Helicopters, Academic Press, 1964.

3. Bisplinghoff, R. L., Ashley, H. and Halfman, R. L., Aeroelasticity, Addison-Wesley, 1955.

4. Simon, K.R., Mechanics, Addison-Wesley, 1960.

5. Hurty, W. C., and Rubinstein, M. F., Dynamics of Structures, Prentice-Hall, 1964.

6. Yntema, R.T., Simplified Procedures and Charts for the Rapid Estimation of Bending Frequencies of Rotating Beams, NACA TN 3459, June 1955.

7. Wheatley, J. B., An Aerodynamic Analysis of the Autogiro with a Comparison Between Calculated and Experimental Results, NACA Report No. 498, 1934.

8. Bailey, F. J., A Simplified Theoretical Method of Determining the Characteristics of a Lifting Rotor in Forward Flight, NACA Report No. 716, 1941.

9. Etkin, B., Dynamics of Flight, John Wiley & Sons, 1959.

10. Critzos, C. C., Heyson, H. H., and Boswinkle, R. W. Jr., Aerodynamic Characteristics of NACA 0012 Airfoil Section At Angles of Attack from 0° to 180°, NACA TN 3361

11. Hoerner, S. F., Fluid Dynamic Drag, published by the author, 1965.